Advances on computational imaging, material appearance, and virtual reality

Ana Belén Serrano Pacheu

CESA
CONSEJO ECONOMICO
Y SOCIAL DE ARAGÓN

CONSEJO ECONÓMICO Y SOCIAL DE ARAGÓN
COLECCIÓN TESIS DOCTORALES

Premio a trabajos de investigación o tesis doctorales 2019

Autor de la tesis doctoral:
Ana Belén Serrano Pacheu

Directores de la tesis:
Belén Masia
Diego Gutierrez

Calificación obtenida:
Sobresaliente *cum laude*

La responsabilidad por las opiniones expresadas en las publicaciones del Consejo Económico y Social de Aragón incumbe exclusivamente a sus autores y su publicación no significa que el Consejo se identifique con las mismas.

2020

Esta publicación se edita únicamente en formato digital.

Consejo Económico y Social de Aragón
c/ Joaquín Costa, 18, 1º
50071 Zaragoza (España)
Teléfono: 976 71 38 38 – Fax: 976 71 38 41
cesa@aragon.es
www.aragon.es/cesa

Premio del Consejo Económico y Social de Aragón 2019 a trabajos de investigación concluidos o tesis doctorales

El Consejo Económico y Social de Aragón, con el fin de promover y divulgar la investigación, convoca anualmente Premios a trabajos de investigación concluidos o tesis doctorales, en cuya convocatoria del año 2019, efectuada por Resolución de 3 de junio de 2019, de la Presidencia del Consejo Económico y Social de Aragón (*Boletín Oficial de Aragón* número 111, de 11 de junio de 2019), pudieron participar los autores de trabajos de investigación concluidos o tesis doctorales presentadas para la colación del grado de doctor leídas y calificadas de sobresaliente *cum laude*, por unanimidad, entre el 16 de septiembre de 2018 y el 15 de septiembre de 2019.

Por Resolución de 10 de diciembre de 2019, de la Secretaría General Técnica de la Presidencia (*Boletín Oficial de Aragón* número 244, de 17 de diciembre de 2019), se otorgó el premio del Consejo Económico y Social de Aragón 2019 a trabajos de investigación concluidos o tesis doctorales.

El premio, con una dotación económica de 3.000 euros, se otorgó a la tesis doctoral *Advances on computational imaging, material appearance, and virtual reality*, realizada por Dª. Ana Belén Serrano Pacheu.

La Comisión de Valoración, que propuso por unanimidad la concesión del Premio, estuvo compuesta por los siguientes miembros del Consejo:

Presidente:	D. José Manuel Lasierra Esteban
Secretaria:	Dª. Belén López Aldea
Vocales:	D.ª Mª. Dolores Gadea Rivas
	Dª. Sonia García Fabregat
	D.ª Beatriz Callén Escartín

Abstract / Resumen

Visual computing is a recently coined term that embraces many subfields in computer science related to the acquisition, analysis, or synthesis of visual data through the use of computer resources. What brings all these fields together is that they are all related to the visual aspects of computing, and more importantly, that during the last years they have started to share similar goals and methods. This thesis presents contributions in three different areas within the field of visual computing: computational imaging, material appearance, and virtual reality.

The first part of this thesis is devoted to computational imaging, and in particular to rich image and video acquisition. First, we deal with the capture of high dynamic range images in a single shot, where we propose a novel reconstruction algorithm based on sparse coding and reconstruction to recover the full range of luminances of the scene being captured from a single coded low dynamic range image. Second, we focus on the temporal domain, where we propose to capture high speed videos via a novel reconstruction algorithm, again based on sparse coding, that allows recovering high speed video sequences from a single photograph with encoded temporal information.

The second part attempts to address the long-standing problem of visual perception and editing of real world materials. We propose an intuitive, perceptually based editing space for captured data. We derive a set of meaningful attributes for describing appearance, and we build a control space based on these attributes by means of a large scale user study. Finally, we propose a series of applications for this space. One of these applications to which we devote particular attention is gamut mapping. The range of appearances displayable on a particular display or printer is called the gamut. Given a desired appearance, that may lie outside of that gamut, the process of gamut mapping consists on making it displayable without excessively distorting the final perceived appearance. For this task, we make use of our previously derived perceptually-based space to introduce visual perception in the mapping process to help minimize the perceived visual distortions that may arise during the mapping process.

The third part is devoted to virtual reality. We first focus on the study of human gaze behavior in static omnistereo panoramas. We collect gaze samples and we provide an analysis of this data, proposing then a series of applications that make use of our derived insights. Then, we investigate more intricate behaviors in dynamic environments under a cinematographic context. We gather gaze data from viewers watching virtual reality videos containing different edits with varying parameters, and provide the first systematic analysis of viewers' behavior and the perception of continuity in virtual reality video. Finally, we propose a novel method for adding parallax for 360° video visualization in virtual reality headsets.

Esta tesis se centra en tres campos relacionados con la computación visual (*visual computing*): imagen computacional, apariencia de materiales, y realidad virtual.

La primera parte de la tesis se centra en el campo de la imagen computacional (*computational imaging*), y en particular, la captura de imágenes de alto rango dinámico, y de vídeo de alta velocidad. En el primer caso, nos centramos en la captura de imágenes de alto rango dinámico en un solo disparo. Proponemos un novedoso algoritmo de reconstrucción basado en *sparse coding* que nos permite reconstruir el rango completo de luminancias de una escena, a partir de una sola imagen en bajo rango dinámico con la información de luminancia codificada. En el segundo caso, nos centramos en el dominio temporal, proponiendo un algoritmo de reconstrucción que nos permite recuperar secuencias de video de alta velocidad a partir de una sola imagen capturada, en la que la información temporal se encuentra codificada.

La segunda parte de la tesis se centra en percepción visual y edición de apariencia de materiales del mundo real. En este trabajo, proponemos un espacio intuitivo y basado en percepción para la manipulación y edición de materiales capturados. Para ello, derivamos una serie de atributos significativos para describir apariencia, y después construimos un espacio para manipular dichos atributos basándonos en un estudio de usuarios a gran escala. Después, proponemos una serie de aplicaciones que pueden derivarse del uso de este espacio. Una de estas aplicaciones, a la que dedicamos especial atención, es el proceso de *mapeado de tono* (*gamut mapping*). El rango de apariencias que los dispositivos de reproducción (e.g., pantallas o impresoras) pueden mostrar está limitado, y es diferente para cada dispositivo. Dada una apariencia deseada, que puede estar fuera de esta gama que los dispositivos pueden mostrar, el proceso de *mapeado de tono* consiste en convertir dicha apariencia a una que el dispositivo sí que pueda representar, intentando minimizar las alteraciones visuales que le podamos causar en este proceso. Para esta tarea, hacemos uso de nuestro espacio de manipulación basado en atributos perceptuales para introducir percepción visual en este proceso, y así intentar minimizar las distorsiones visuales percibidas que puedan surgir durante el *mapeado de tono*.

La tercera parte de la tesis se centra en realidad virtual. Primero, nos centramos en el estudio del comportamiento y la mirada de los usuarios en panoramas estáticos con estéreo en todo el campo de visión (*omnistereo*). Para ello, capturamos y analizamos muestras de varios usuarios, y además proponemos una serie de aplicaciones haciendo uso de nuestras conclusiones. Después, investigamos comportamientos más complejos, esta vez en entornos dinámicos bajo un contexto cinematográfico. Para ello, capturamos datos de la mirada de los usuarios visualizando vídeos en realidad virtual que contienen cortes con diferentes parámetros, y con estos datos realizamos el primer análisis sistemático de la percepción de continuidad y el comportamiento de los usuarios visualizando contenido cinematográfico en realidad virtual. Finalmente, proponemos un nuevo método para añadir paralaje a la visualización en tiempo real de vídeos en realidad virtual. Este método permite que vídeos que habían sido capturados desde un único punto de vista, puedan ser visualizados desde puntos de vista ligeramente diferentes, mejorando la inmersión y la calidad de la experiencia virtual.

Measurable contributions and merits

The realization of this thesis has led to the following results, which are detailed in Section 1.5.1:

- 9 JCR-indexed journal publications (including 2 in ACM Transactions on Graphics, and 2 in IEEE Transactions on Visualization and Computer Graphics) [205, 215, 289, 290, 292–294, 298, 311]
- 3 peer-reviewed conference publications [8, 204, 288]
- 4 posters and workshops [212, 287, 291, 310]
- 1 patent (pending resolution) [81]
- 2 research stays (totaling four months) at Stanford University (California, United States)
- 2 research stays (totaling seven months) at Max Planck Institute for Informatics (Saarbrücken, Germany)
- 1 research stay (three months) at Adobe Research (California, United States)
- 1 PhD grant, 2 mobility grants, a Nvidia Graduate Fellowship Award, an Adobe Research Fellowship Honorable Mention, a 1st place at the ACM Student Research Competition (undergraduate category), and a *Tercer Milenio* young research talent award
- 8 supervised students of a 4-year degree at their final projects (Trabajo Fin de Grado)
- 1 supervised Master thesis
- 2 best paper awards, and 1 honorable mention
- Participation in 4 research projects
- Reviewer for 8 journals, 5 international conferences, and Program Committee Member for two international conferences and one national conference

Acknowledgments

During these years many people have helped me in one way or another, and this thesis would not be complete without extending all my gratitude to them, so to everyone that has been there to support me at some point along the way: thank you. In particular, I would like to thank a series of people.

Diego, I had no idea of what I wanted to do with my career until I attended your classes. Thank you for showing me the magic of computer graphics, and for giving me the chance of pursuing this PhD. Also, for giving me perspective, and teaching me to look at the big picture.

Belen, for so many things. Thanks for helping me to be a better researcher, for your patience to my countless knocks in your door, and for everything you have taught me. For all the good advices, and all your help during these years.

The members *Graphics and Imaging Lab* for being a great group. For discussing tons of ideas, sharing all their knowledge, and for their invaluable help, specially during deadlines.

The students I have supervised, *Nicolás Landa, Marcos Allué, Santiago Calvo, Jaime Ruiz-Borau, Sandra Malpica, Miguel Barrio, Javier Camón,* and *Miguel Martinez*. Thanks for your trust, I really hope you enjoyed the time spent in your projects, as I enjoyed supervising you.

Karol Myszkowski not only for the months hosting me, but also for all the ideas discussed, and all the support during my stay and after.

Gordon Wetzstein for hosting me both at the beginning of this PhD, when I still had no idea of what I was doing, and at later stages. For all the good advices, and enthusiasm you shared with me.

My hosts at Adobe Research, *Stephen DiVerdi and Aaron Hertzmann*. Thank you for giving me the opportunity of working in such a nice environment, and for being understanding when I needed it.

All the collaborators I have worked with during this years. Thanks for sharing your knowledge, and for the hard work to make our projects successful.

My family and friends, for being there for me, and for pushing me to stop working from time to time and have fun (even though it was rarely successful).

My parents *Carlos* and *Ana* for their love, and for teaching me so many things. For showing me that working hard pays off, and for encouraging me to follow the path I wanted to follow. For being there when I needed the most.

This work has been funded by the European Research Council (project CHAME-LEON), the Spanish Ministry of Economy and Competitiveness (projects TIN2013-41857-P, TIN2014-61696-EXP, TIN2016-79710-P, and TIN2016-78753-P), the Max Planck Center for Visual Computing and Communication, Adobe Research, and the Nvidia's Graduate Fellowship program.

Índice

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Part I

Introduction and overview

1. Introduction

Visual computing is a recently coined term that embraces many subfields in computer science related to the acquisition, analysis, or synthesis of visual data through the use of computer resources [305]. What brings all these fields together is that they are all related to the visual aspects of computing, and more importantly, that during the last years they have started to share similar goals and methods. This includes fields such as computational imaging (to capture images), computer vision (to derive information from these images), rendering (to generate visual representations from virtual models), modeling the properties of real-world objects (such as appearance or function), visualization (including both traditional displays and virtual/augmented reality), and cognitive sciences (such as perception or human-computer interaction, to bridge the virtual and real worlds).

This thesis presents contributions to three different areas within the field of visual computing: computational imaging (Part ii), material appearance (Part iii), and virtual reality visualization (Part iv). This chapter provides a brief introduction into these three areas, and the contributions of this thesis to each of them.

1.1 Computational imaging

The first part of this thesis focuses on the field of computational imaging, with the goal of exploring alternative capture methods for rich image acquisition. Adelson and Bergen [2] defined the plenoptic function as a complete representation of the visual world, a description of every possible photograph that can be taken of a particular space-time slice of a scene (see Figure 1.1 left for a graphic representation). This is formalized as light being a function of position (x, y, z), direction (θ, ϕ), wavelength λ, and time t: $L = L(x, y, z, \theta, \phi, \lambda, t)$. Given the difficulty of sampling all these dimensions at the same time, traditional imaging typically samples only a subset of the dimensions of this space, integrating the rest of dimensions and thus losing an enormous amount of information in the process. For example, a conventional photograph will sample three dimensions (x, y, λ) while integrating the angular and temporal domains. Computational imaging [124, 260] aims to overcome the limitations of traditional imaging systems, allowing for richer representations of the scene by combining optics, specialized hardware, and computation. One of the key ideas used is coding additional information of the scene using different multiplexing techniques (e.g., coded sensor or illumination) to augment the captured data, and then recover all this information after the capture by means of computation.

There is a wide body of work devoted to capturing the different dimensions of the plenoptic function more efficiently (see the work of Wetzstein et al. [339] for a survey). Examples of this include cameras that sample the angular dimension [214, 231], and multispectral cameras that sample the spectral dimension more densely [18, 62, 153] by coding the light at the sensor, cameras that allow for correcting defocus blur by coding the aperture [217, 359], or cameras that can capture sharp images from moving scenes by coding the shutter [261]. Recently, it has been demonstrated that light can be captured at a temporal resolution of picoseconds, allowing us to see light propagating through a macroscopic scene [328].

In particular, this thesis makes contributions to (i) efficiently capture the temporal dimension t of the plenoptic function, for which temporal information at different time instants is captured in a single image by coding the light that reaches the sensor, and later the full video sequence is reconstructed by using this coded information, and (ii) faithfully capturing the luminance ranges L of this function (high dynamic range imaging), for which the amount of light that arrives to

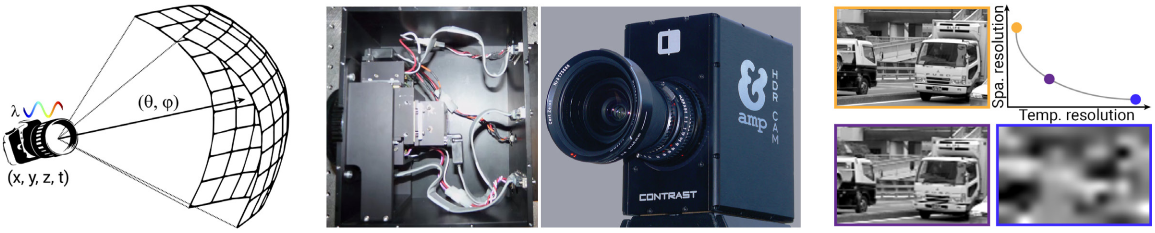


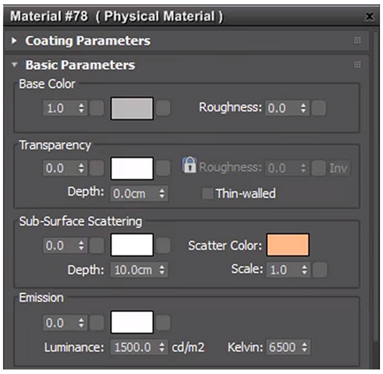
Figure 1.1: Left: Plenoptic function. Light is represented as a function of position (x, y, z), direction (θ, ϕ), wavelength λ, and time t. Center: Depiction of the camera built by Tocci et al, [319], which uses beamsplitters and three sensors for HDR image capture. Right: Representation of the tradeoff between spatial and temporal resolution in video capture.

the sensor is captured with different per-pixel exposures, and later the full dynamic range is reconstructed.

First, video capture is limited by the trade-off between spatial and temporal resolution: when capturing videos at high temporal resolution, the spatial resolution decreases due to bandwidth limitations in the capture system (see Figure 1.1 right). Achieving both high spatial and temporal resolution is only possible with highly specialized and very expensive hardware, and even then the same basic trade-off remains. Our technique allows for the capture of single-shot high-speed video by coding the temporal information in a single frame, and then reconstructing the full video sequence from this single coded image.

Second, current high dynamic range (HDR) acquisition techniques are based on either (i) fusing multibracketed, low dynamic range (LDR) images, (ii) modifying existing hardware and capturing different exposures simultaneously with multiple sensors (see for example Figure 1.1 center), or (iii) reconstructing the full range of luminances from a single image with spatially-varying pixel exposures. Our approach relies on the latter, requiring only minimal hardware modifications. However, it can also be used with commercial cameras capable of capturing interleaved exposures. We have proposed a novel algorithm to recover high-quality HDR images from a single, coded LDR image that achieves large dynamic ranges without trading off resolution. Our approach is single-shot, thus removing the need for alignment, motion estimation or, in general, any deghosting strategy. These features allow for naturally extending our method to HDR video capture without enforcing additional constraints.

Both our contributions, high-speed video capture and single-shot HDR imaging, rely on *sparse coding* and the recent theories of *compressed sensing*, and our reconstruction algorithms are based on *convolutional sparse coding* (CSC) techniques. *Sparse coding* is the operation of finding a sparse representation of a given signal in an alternative domain. *Compressed sensing* is a signal processing technique for efficiently acquiring and reconstructing a signal, based on the principle that this sparsity of a signal in an alternative domain can be exploited to fully recover it from fewer samples than required by the Shannon-Nyquist theorem. This allows for coding, within a single image, additional information that can be later reconstructed. *Compressed sensing* is closely related to *sparse coding*, however it is specifically targeted towards the goal of finding the sparsest solution to an *under-determined* problem, while the term *sparse coding* refers to a more general case (it does not necessarily deal with an under-determined system). For the reconstruction step, we formulate our optimization algorithms relying on CSC techniques. CSC is a strategy for unsupervised learning of image features: a learned *dictionary* of image features (our alternative domain) can be later used for reconstruction tasks. *Sparse coding* reconstruction techniques assume independence between observations during the learning process, however, they are typically applied in a patch-based manner, where this assumption breaks. CSC is closely related to traditional patch-based methods, however it allows for operating in the complete image without the need of subdividing into patches, thus explicitly modeling local interactions and capturing the correlation between local neighbors.



Material #78 (Physical Material)

Coating Parameters

Basic Parameters

Base Color

1.0 Roughness: 0.0

Transparency

0.0 Roughness: 0.0 Inv

Depth: 0.0cm Thin-walled

Sub-Surface Scattering

0.0 Scatter Color:

Depth: 10.0cm Scale: 1.0

Emission

0.0

Luminance: 1500.0 cd/m2 Kelvin: 6500

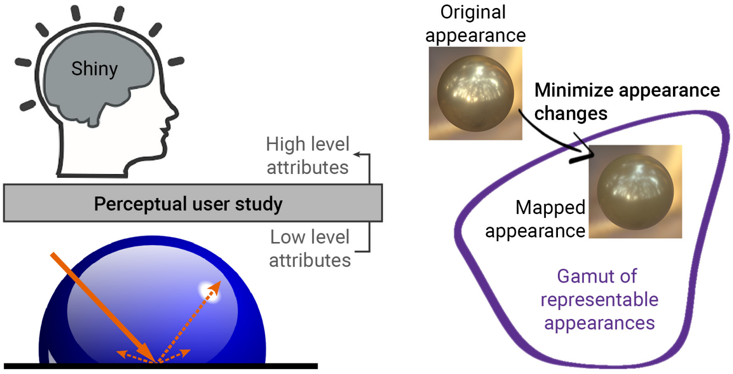


Figure 1.2: Left: A snapshot of a material editing interface currently used in 3D modeling software. The editing process is laborious, and the disconnect between the parameters and how they affect the perceived material make it difficult for the user to convey a desired appearance. Center: We translate low level attributes to high level attributes that are easier to understand for users by means of a large scale user study. Right: We use these high level attributes to derive a similarity metric that can be used in many applications, for example, as a guidance for minimizing appearance changes in the gamut mapping process.

1.2 Material appearance

Simulation and editing of visual appearance is a core area in Computer Graphics. Computer-generated imagery is present in many aspects of our daily life, such as games, movies, architecture, engineering, advertisements, or virtual prototyping. Developing proper design and editing algorithms for visual appearance is not only a fundamental aspect of digital content creation, but also a key feature for the success of novel fields such as computational materials or fabrication. However, editing the visual appearance of computer-generated objects is a challenging goal, requiring careful modeling and adjustment of many parameters, and the complex interactions between them (see Figure 1.2 left for an example of a currently used editor in 3D modeling software). Despite many research efforts, this process remains unintuitive, inefficient and very limited, even for trained practitioners.

The problem is further aggravated by the current data-driven paradigm shift: Measurement techniques for material appearance are gaining in accuracy, efficiency, and ease of use (see [5, 7, 17, 228, 236] for recent examples). Although this has allowed us to improve the level of realism in visual appearance, editing this captured data remains a challenge. First, there is a disconnect between the data representation and any meaningful parameters that humans understand; the captured data typically consists on huge lists of tabulated data, which is not human-friendly, and thus very difficult to handle. Second, the many different acquisition systems lead to heterogeneous formats, which require different editing approaches [87]. And third, real-world appearance functions are usually very high-dimensional, so editing parameters are not intuitive. As a result, visual appearance datasets are increasingly unfit to editing operations, which limits the creative process for scientists, artists, and practitioners in general.

In this thesis we make contributions towards an intuitive, perceptually based editing space for captured data. We depart from existing large tabulated databases of measured reflectance functions, and our goal is to provide an intuitive representation for this data (see Figure 1.2 center). In order to do so, we perform a series of user studies, and we use the collected perceptual data to learn how to map low level attributes of captured materials, to high level intuitive attributes. This mapping can be adapted for several representation formats, and builds on top of human perception in order to provide an intuitive editing space for appearance. We also show that this space can be of use for several applications such as material editing, or deriving similarity metrics.

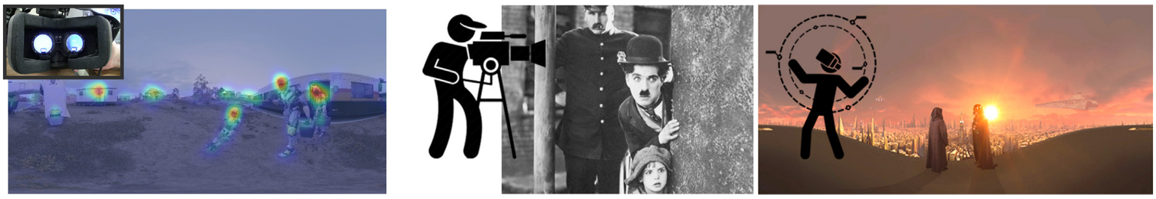


Figure 1.3: Left: We use an eyetracker to capture gaze from users and analyze their behavior in immersive VR environments. The computed heatmap indicates the areas of the panorama to where users payed more attention (salient regions). Right: One of the key aspects that makes VR different from visualizing content in traditional displays, is that now the user controls the camera, and decides in which direction to look.

One of these applications to which we devote particular attention is gamut mapping. Displays and printing devices are limited in the set of appearances that they can represent [304]. Given a desired appearance to represent, and a set of available inks (for the case of printers), the gamut mapping process consists on trying to find a combination of the available inks that will resemble as close as possible the desired appearance (see Figure 1.2 right). This is an extremely underconstrained problem without a unique solution, for which several methods have been proposed [221, 254]. The key problem that makes this process hard to carry out is the lack of a compelling metric to define similarity in terms of appearance. Several metrics have been proposed [104], but typically they do not take into account human perception, which results in very noticeable changes in appearance. We, instead, introduce human perception in the mapping process by making use of our previously derived intuitive space in order to minimize the alteration of the perceived visual appearance.

1.3 Virtual reality

Virtual reality (VR) is a fast-growing medium still in an exploratory phase, with many unknowns, and very exciting avenues for research. With the proliferation of low-cost, consumer level head-mounted displays (HMDs) [125, 240] and capture devices [237, 273], VR is progressively entering the consumer market. VR systems provide a new way of experiencing virtual content that is richer than traditional displays, yet also different from how we experience the real world. These unprecedented capabilities for experiencing new content have the potential to profoundly impact our society.

A key aspect that makes VR different from visualizing content in traditional displays, is that now the user controls the camera, and decides in which direction to look. However, little is known about how this new scenario may affect users' behavior: How does one design or edit virtual environments effectively in order to retain or guide users' attention? Can we predict users' behavior and react accordingly? How does one create a satisfactory cinematic VR experience? On a more fundamental level, our understanding of image perception may have to be revised for VR. To derive conventions for VR video from first principles, it is crucial to understand how users explore virtual environments, and what constitutes attention. Such an understanding would be of use not only for content generation, but would also inform future designs of user interfaces, eye tracking technology, and other key aspects of VR systems. The joint study of cognitive mechanisms and computational techniques provides a solid ground to tackle some of the current challenges.

Our research in this area is targeted towards (i) understanding viewers' behavior in immersive VR environments, (ii) analyzing the mechanisms that constitute attention, and (ii) improving the visual experience in the real-time playback of virtual reality videos. To further our understanding of viewing behavior, first we have started by grounding fundamental knowledge by analyzing users' interaction with simple stimuli. We have captured and analyzed gaze and head orientation data of users exploring stereoscopic, *static* omnidirectional panoramas under different viewing

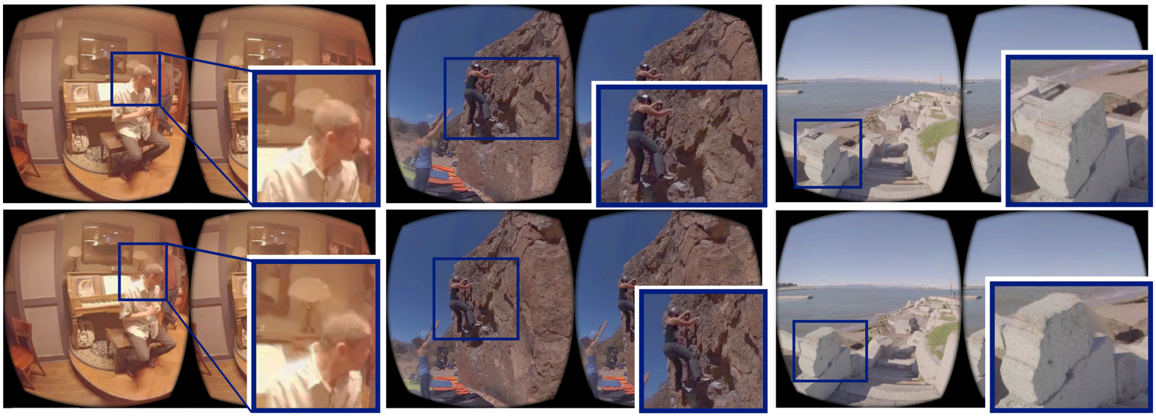


Figure 1.4: Three examples of novel views generated with our method. For each example we show the original view (top), and the corresponding displaced view (bottom). We also include close-ups of regions where the added parallax is clearly visible.

conditions (Figure 1.3 left). We have provided a thorough analysis of this data, which has led to several important insights, such as the existence of a particular fixation bias. In addition, we have explored other applications of our data and analysis, including: automatic alignment of VR video cuts, panorama thumbnails, panorama video synopsis, and saliency-based compression.

Second, we have built on top of that knowledge by analyzing more intricate behaviors in complex environments under a cinematographic context. Traditional cinematography has relied for over a century on a well-established set of editing rules, called continuity editing, to create a sense of situational continuity. Despite massive changes in visual content across cuts, viewers in general experience no trouble perceiving the discontinuous flow of information as a coherent set of events. However, virtual reality movies are intrinsically different from traditional movies in that the viewer controls the camera orientation at all times (Figure 1.3 right). We have investigated key relevant questions to understand how well traditional movie editing carries over to VR, and we have provided the first systematic analysis on perceived continuity in VR. From this analysis we have proposed a number of metrics to describe viewers attentional behavior in VR, and we have made a series of relevant findings. Our final goal is to apply this knowledge to derive new conventions for VR content generation.

Third, we have presented a method for adding parallax for virtual reality 360° video visualization (Figure 1.4). In current video players, the playback does not respond to translational head movement, which reduces the feeling of immersion, and causes motion sickness for some viewers. Given a video and its corresponding depth, a naive image-based rendering approach would use the depth to generate a 3D mesh around the viewer, then translate it appropriately as the viewer moves their head. However, this approach breaks at depth discontinuities, showing visible distortions, whereas cutting the mesh at such discontinuities leads to ragged silhouettes and holes at disocclusions. We have addressed these issues by improving the given initial depth map to yield cleaner, more natural silhouettes. We rely on a three-layer scene representation, made up of a foreground layer and two static background layers, to handle disocclusions by propagating information from multiple frames for the first background layer, and then inpainting for the second one. Our system works with input from many of today's most popular 360° stereo capture devices (e.g., Yi Halo or GoPro Odyssey), and works well even if the original video does not provide depth information. Our user studies have confirmed that our method provides a more compelling viewing experience than without parallax, increasing immersion while reducing discomfort and nausea.

1.4 Goal and overview

This thesis is divided in three main parts, one for each of the three topics described in the introduction.

- Part ii is devoted to computational imaging. In Chapter 2 we deal with the capture of high dynamic range images in a single shot, proposing a novel reconstruction algorithm based on convolutional sparse coding to recover the full range of luminances from a single coded low dynamic range image. Chapter 3 deals with the temporal dimension, proposing a novel reconstruction algorithm, again based on convolutional sparse coding, to recover high-speed video sequences from a single coded image.
- Part iii tackles the long-standing problem of visual perception and editing of real world materials. In Chapter 4 we propose an intuitive, perceptually based editing space for captured data. We derive a set of meaningful attributes for describing appearance, and we build a control space based on these attributes by means of a large scale user study. Finally, we propose a series of applications for this space. One of these applications, to which we devote particular attention in Chapter 5, is gamut mapping. We make use our space to introduce human perception in the mapping process to help choosing the appropriate inks to represent the desired appearance.
- Part iv is devoted to virtual reality. In Chapter 6 we first collect gaze samples and provide the first dataset of gaze behavior in static omnistereo panoramas. Then, we provide an analysis of this data, and propose a series of applications that make use of our derived insights. In Chapter 7 we investigate more intricate behaviors in dynamic environments under a cinematographic context. We gather gaze data from viewers watching VR videos containing different edits with varying parameters, and provide the first systematic analysis of viewers' behavior and the perception of continuity in VR. In Chapter 8 we present a method for adding parallax for virtual reality 360° video visualization. Our user studies confirm that our method provides a more compelling viewing experience than without parallax, increasing immersion while reducing discomfort and nausea.

While I am the leading author in many of the works presented in this thesis, they have been done in collaboration with different colleagues. To favor readability, the work described is contextualized at the beginning of each chapter and, when needed, my contribution is explicitly described.

1.5 Contributions and measurable results

1.5.1 *Publications*

A large amount of the work presented in this thesis has been published in seven JCR indexed journals (including two papers in ACM Transactions on Graphics presented at SIGGRAPH and SIGGRAPH Asia, and 2 papers in IEEE Transactions on Visualization and Computer Graphics, presented at IEEE VR), and two peer-reviewed conference publications:

- Convolutional Sparse Coding for High Dynamic Range Imaging (Chapter 2, Part ii)
  - This work was accepted to Eurographics (EG) 2016, and published in Computer Graphics Forum [290]. This journal has an impact factor of 1.611, and its position in the JCR index is 44th out of 106 (Q2) in the category Computer Science, Software Engineering (data from 2016).
- Convolutional sparse coding for capturing high-speed video content (Chapter 3, Part ii)

  - This work is published in Computer Graphics Forum [294]. This journal has an impact factor of 2.046, and its position in the JCR index is 22th out of 104 (Q1) in the category Computer Science, Software Engineering (data from 2017).
  - Previous results were published in the national conference Congreso Español de Informática Gráfica (CEIG) 2015 [288].
  - Preliminary analyses were presented as a poster at ICCP 2015 [287].

- An intuitive control space for material appearance (Chapter 4, Part iii)
  - The main work was accepted to SIGGRAPH Asia 2016, and published in ACM Transactions on Graphics [289]. This journal has an impact factor of 4.088, and its position in the JCR index is 1st out of 106 (Q1) in the category Computer Science, Software Engineering (data from 2016).
  - Previous results were presented as a poster at SIGGRAPH 2016 [291].
  - An exploration of potential improvements to the presented control space (Section 4.9) was presented in the national conference Congreso Español de Informática Gráfica (CEIG) 2017 [204].
  - A brief overview of current challenges in intuitive editing of visual appearance was presented in the Material Appearance Modeling Workshop 2017 [212].

- Attribute-preserving gamut mapping of measured BRDFs (Chapter 5, Part iii)
  - This work was accepted to the Eurographics Symposium on Rendering (EGSR) 2017, and published in Computer Graphics Forum [311]. This journal has an impact factor of 2.046, and its position in the JCR index is 22th out of 104 (Q1) in the category Computer Science, Software Engineering (data from 2017).
  - Previous results were presented as a poster at SIGGRAPH 2017 [310].

- Saliency in VR: How do people explore virtual environments? (Chapter 6, Part iv)
  - The main work was accepted to IEEE VR 2018, and published in IEEE Transactions on Visualization and Computer Graphics [298]. This journal has an impact factor of 3.078, and its position in the JCR index is 8th out of 104 (Q1) in the category Computer Science, Software Engineering (data from 2017).

- Movie Editing and Cognitive Event Segmentation in Virtual Reality Video (Chapter 7, Part iv)
  - The main work was accepted to SIGGRAPH 2017, and published in ACM Transactions on Graphics [292]. This journal has an impact factor of 4.384, and its position in the JCR index is 3rd out of 104 (Q1) in the category Computer Science, Software Engineering (data from 2017).

- Motion parallax for 360 RGBD video (Chapter 8, Part iv)
  - The main work was accepted to IEEE VR 2019, and will be published in IEEE Transactions on Visualization and Computer Graphics [293]. This journal has an impact factor of 3.078, and its position in the JCR index is 8th out of 104 (Q1) in the category Computer Science, Software Engineering (data from 2017).

In addition to these previous publications, during my PhD I have participated in other research projects not directly related with the topic of this thesis:

- *Dynamic range expansion based on image statistics*. The topic of this work, led by Belen Masia, is reverse tone mapping, or how to expand the dynamic range of a conventional image to be shown on a high dynamic range display; the main contribution is that further explore automatic mappings based on image statistics. It has been published in Multimedia Tools and Applications [215]. This journal has an impact factor of 1.541, and its position in the JCR index is 42th out of 104 (Q2) in the category Computer Science, Software Engineering (data from 2017).

- *Crossmodal perception in immersive environments*. In this work we analyzed the crossmodal interactions between visual and auditory stimuli in virtual reality by replicating in a head mounted display a well-known crossmodal perception experiment previously carried out in conventional displays. It has been presented in the national conference Congreso Español de Informática Gráfica (CEIG) 2016 [8].
- *Crossmodal Perception in Virtual Reality*. This work is an extension to journal of a previous work presented in the national conference Congreso Español de Informática Gráfica (CEIG) [8]. In this work we explore more complex crossmodal interactions between visual and auditory stimuli in virtual reality when judging material appearance. It has been accepted to the journal Multimedia Tools and Applications [205]. This journal has an impact factor of 1.541, and its position in the JCR index is 42th out of 104 (Q2) in the category Computer Science, Software Engineering (data from 2017).

1.5.2 *Awards*

This subsection includes a list of awards and fellowships received throughout this thesis. Their generous support has allowed to carry out the work here presented:

- FPI grant from the Spanish Ministry of Economy and Competitiveness (4-year PhD grant).
- Nvidia Graduate Fellowship Award (including a donation of $50000).
- Adobe Research Fellowship Honorable Mention (including a donation of $2000 and a Creative Cloud membership).
- *Tercer Milenio* award (young research talent category), granted by the *Heraldo de Aragon* newspaper.
- Two mobility grants from the Spanish Ministry of Economy and Competitiveness (4-month grant for carrying out short stays in research centers).

Additionally, some of the projects described in this thesis have been awarded a special recognition:

- 1st place at the ACM Student Research Competition Grand Final (undergraduate category) for the work *Attribute-preserving gamut mapping of measured BRDFs* [311].
- Best paper Honorable Mention at Eurographics 2016 for the work *Convolutional sparse coding for high dynamic range imaging* [290].
- Best paper (1 in 2) at the national conference Congreso Español de Informática Gráfica (CEIG) 2016 for the work *Crossmodal perception in immersive environments* [8] (proposed for extension and submission to the journal Computer Graphics Forum).
- Best paper (1 in 2) at the national conference Congreso Español de Informática Gráfica (CEIG) 2015 for the work *Compressive high-speed video acquisition* [288] (proposed for extension and submission to the journal Computer Graphics Forum).

1.5.3 *Research stays*

Four research stays and internships were carried out at different institutions during the course of this thesis, totaling 12 months:

- September 2018 – November 2018 (two months): Visiting Student at the Max-Planck Computer Graphics Group, Max-Planck-Institut für Informatik. Supervisor: Prof. Dr. Karol Myszkowski.
- June 2017 - August 2017 (three months): Research Intern at the Creative Technologies Lab, Adobe Research. Supervisors: Dr. Stephen DiVerdi and Dr. Aaron Hertzmann.
- June 2016 - August 2016 (three months): Visiting Student at the Stanford Computational Imaging Group, Stanford University. Supervisor: Prof. Dr. Gordon Wetzstein.
- August 2015 – January 2016 (five months): Visiting Student at the Max-Planck Computer Graphics Group, Max-Planck-Institut für Informatik. Supervisor: Prof. Dr. Karol Myszkowski.
- April 2015 – May 2015 (one month): Visiting Student at the Stanford Computational Imaging Group, Stanford University. Supervisor: Prof. Dr. Gordon Wetzstein.

1.5.4 *Supervised students*

Throughout this thesis I have supervised the final degree project (Spanish 4-year engineering degrees) of the following students:

- 2018: Miguel Martinez. *3D scene modeling for VR video techniques benchmarking*. Grade: 9.5/10.0 - with honors.
- 2017: Javier Camón. *Editing in VR: influence of sound techniques on narrative continuity*. Grade: 9.5/10.0 - with honors.
- 2017: Miguel Barrio. *Improved intuitive spaces for material appearance editing*. Grade: 9.0/10.0.
- 2017: Sandra Malpica. *Improvement of a material representation model for intuitive appearance editing*. Grade 9.4/10.0.
- 2017: Jaime Ruiz-Borau. *Narrative continuity in virtual reality*. Grade: 8.5/10.0.
- 2016: Santiago Calvo. *Content-aware texture and style transfer using convolutional neural networks*. Grade: 9.2/10.0.
- 2016: Marcos Allué. *Crossmodal perception in immersive environments*. Grade: 9.2/10.0.
- 2015: Nicolás Landa. *Compressive sensing techniques for image acquisition*. Grade: 9.3/10.0.

I have also co-supervised one master thesis:

- 2018: Sandra Malpica. *Study and and applications of human sensory perception in virtual reality*. Grade: 9.5/10.0 - with honors.

1.5.5 *Research projects*

During my PhD studies I have been involved in the following research projects:

- CHAMELEON: Intuitive editing of visual appearance from real-world datasets. *European Research Council (ERC). Grant agreement N°: 682080*. PI: Diego Gutierrez.

- REVEAL: Scene Recovery Using an Extended Plenoptic Function. *Defense Advanced Research Projects Agency (DARPA). Research Subcontract N^o: 678K904*. PI: Diego Gutierrez.

- LIGHTSLICE: Captura, análisis y aplicaciones del transporte de luz multidimensional (Aplicación a imagen médica). *Spanish Ministry of Economy and Competitiveness. Project N^o: TIN2013-41857-P*. PI: Diego Gutierrez.

- IMAGER: Imagen Computacional. Técnicas avanzadas de edición y visualización mediante computación en dominios alternativos. *Spanish Ministry of Economy and Competitiveness. Project N^o: TIN2016-79710-P*. PI: Belen Masia.

Part II

Computational imaging

The first half of this part is devoted to single-shot high dynamic range imaging; the main contribution is the proposed coding and reconstruction algorithm. The topic of the second half is high-speed video acquisition; in particular the main contribution lies in the reconstruction algorithm to recover a temporal sequence from a single coded image.

2. Convolutional sparse coding for high dynamic range imaging

ABOUT THIS CHAPTER

The work presented in this chapter has been presented at *Eurographics* 2016, and published in the journal *Computer Graphics Forum*. While I led the line of work (under the supervision of Diego Gutierrez and Belen Masia), this work was conducted in collaboration with *Stanford university*. In particular, Felix Heide collaborated with the derivation of the method described based on Convolutional Sparse Coding.

A. Serrano, F. Heide, D. Gutierrez, G. Wetzstein, and B. Masia.
CONVOLUTIONAL SPARSE CODING FOR HIGH DYNAMIC RANGE IMAGING.
Computer Graphics Forum, Vol. 35 (2), 2016.

2.1 Introduction

One of the fundamental characteristic of a sensor is its *dynamic range*: the interplay of full-well capacity, noise, and analog to digital conversion. The ability to simultaneously record and distinguish very low signals alongside extremely bright scene parts is critical for many applications in scientific imaging, microscopy, and also consumer photography. Unfortunately, the hardware capabilities of available image sensors are insufficient to capture the wide range of intensities observed in natural scenes. This has motivated researchers to develop computational imaging techniques to overcome the dynamic range constraints of sensor hardware by co-designing image capture mechanisms and post-processing algorithms.

Today, high dynamic range (HDR) photography is well-established and usually done via one of three general approaches: sequentially capturing and subsequently fusing multiple different exposures (e.g., [78, 208]), capturing different exposures simultaneously with multiple sensors (e.g., [319]), or coding per-pixel or per-scanline exposures within a single image with appropriate reconstruction algorithms [120, 138, 176, 229, 230, 338, 357]. Whereas sequential image capture is easily afforded by existing cameras, this method makes it challenging to capture dynamic scenes and usually requires additional motion stabilization and de-ghosting techniques. Multi-sensor solutions are elegant, but more expensive and they require precise calibration. In this work, we advocate for coded pixel exposure techniques and propose a new reconstruction algorithm for this class of computational cameras. Our approach builds on recent advances in convolutional sparse coding and reconstruction techniques. We show that a naïve application of traditional, patch-based (i.e. non-convolutional) sparse reconstruction techniques[48, 184] struggles to deliver high image quality for high contrast scenes. We make the key observation that convolutional sparse coding (CSC) (e.g., [161]), is particularly well-suited for the type of high-contrast signals present in HDR images. Therefore, we pose the HDR recovery problem as convolutional sparse coding problem and derive necessary formulations to solve it efficiently. An example of the results achieved with our method can be seen in Figure 2.1. Code and additional results of our method can be found in the project website[1].

We make the following contributions:

- We introduce convolutional sparse coding (CSC) for high dynamic range image reconstruction.

[1] http://webdiis.unizar.es/~bmasia/pubs/eg2016/project_page_CSCHDR

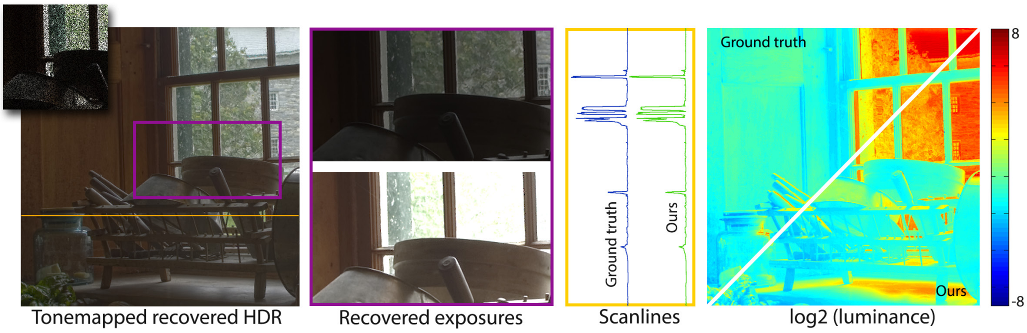


Figure 2.1: High dynamic range image (HDRI) recovered from a single, coded, 8-bit low dynamic range (LDR) image using the proposed sparse reconstruction method. *Left:* HDR image recovered with our framework, tonemapped for display purposes. The inset shows a cropped region of the coded LDR image used as input to the reconstruction algorithm. *Center left:* Close-up of two exposures of the reconstructed HDR image showing the ability of our method to reconstruct an extended dynamic range. *Center right:* normalized luminance plots of the marked scanline (yellow line, rotated by 90°) for the reconstructed image (green curve) and the ground truth image (blue curve). *Right:* false color image of the reconstructed HDR scene (scale is in stops), showing the extremely large dynamic range that the original scene had and our technique is able recover.

- We propose forward and inverse methods that are tailored to recovering a high-contrast (HDR) image from a single, coded exposure photograph.
- We demonstrate improved image quality over other existing approaches and over a naïve application of sparse reconstruction techniques to HDRI. We also evaluate algorithmic parameters, analyze different exposure coding schemes, and interpret HDR image features.
- We build a prototype coded exposure camera and demonstrate the utility of our algorithm using data captured with this prototype.

2.2 Related work

One of the most common techniques to compute HDR images is exposure bracketing [265]. This technique, also known as multi-bracketing, merges several LDR images of the scene taken with different bracketing exposures, into the final HDR image [78, 208]. One of the main drawbacks of this technique is that, if either the camera or some scene elements move during the extended capture process, ghosting artifacts appear. There have been many algorithms designed to remove these artifacts by means of alignment and de-ghosting [301]. Some recent works include the use of optical flow [362], patch-based reconstruction [107, 286], or modeling the noise distribution of color values [116]. The problem is further aggravated for HDR video (e.g., [119, 160, 207]): on the one hand, optical flow solutions fail in the presence of complex motion, on the other hand patch-based methods lack built-in temporal coherence. In contrast, the proposed convolutional sparse coding approach can produce an HDR image from a single shot, thus removing the need for alignment, motion estimation or, in general, any de-ghosting strategy.

Other works rely on multiple cameras [21, 278], enhanced sensor control electronics performed in simulation [257], or otherwise highly modified hardware designs [206, 357]. For instance, Tocci et al. [319] and Kronander et al. [176] achieve single-shot HDR by acquiring several LDR images with different sensors using a beam splitter. Our method uses an off-the-shelf camera with a simple mask on the sensor or using a per-pixel coding exposure, which greatly reduces complexity, size and overall cost.

Previously proposed single-shot approaches rely on exposures that vary per image scanline, for example implemented with coded electronic shutters [61], or sensors which allow different gain settings simultaneously for alternating pixel rows [120, 128, 138]. In all of these cases, an image is reconstructed using sophisticated interpolation methods, and often relies on additional image priors. These methods present a trade-off between the dynamic range that can be recovered with only two different exposures, and the quality of the final reconstruction, determined by how far apart the exposures are chosen. Other spatially-varying gain methods aim at capturing increased dynamic range from a single image, using a per-pixel coded exposures. Nayar and colleagues [229, 230] place a mask of spatially varying neutral density filters on the sensor, effectively coding different exposures for adjacent pixels according to the optical pattern of the mask. However, this method is limited by interpolation artifacts and aliasing resulting from the regular pattern of the mask. The work by Aguerrebere and colleagues [3] leverages recent advances in solving inverse problems [346] together with a spatially-varying mask, but still relies on a complex MAP Expectation-Maximization optimization framework which can lead to artifacts in scenes of high dynamic range.

In this work, we propose a sparse reconstruction framework that takes advantage of the compressibility of visual information to reconstruct a high dynamic range image from a single shot with pixel-coded exposure. Sparse reconstruction has been used before in the context of rendering [284], and image reconstruction and acquisition [226, 282], including high-speed video [195, 288], dual photography [283, 285] and light transport acquisition [251], light field capture [214], hyperspectral imaging [153, 192], or even extended dynamic range imaging using a Fourier basis [276, 277]. However, we do not rely on a conventional, patch-based learning and reconstruction method as most of these works do because it has certain limitations for the recovery of HDR images. Instead, we propose a novel formulation based on convolutional sparse coding (CSC). CSC has been used for learning hierarchical image representations [57, 161, 353] and to solve transient imaging problems [139, 143]. We build on the basic idea of convolutional sparse coding and make it practical for coded, single-shot HDR image acquisition.

2.3 CSC framework for HDR reconstruction

In this section, we offer a brief review of sparse coding techniques and introduce a new formulation of convolutional sparse coding tailored to the problem of high dynamic image reconstruction from a single image with spatially-varying pixel exposures.

2.3.1 *Review of sparse coding and reconstruction*

The traditional problem faced in sparse reconstruction is that of solving an underdetermined system of linear equations $\mathbf{y} = \mathbf{\Phi}\boldsymbol{\alpha}$ in which $\boldsymbol{\alpha} \in \mathbb{R}^n$ is the signal we are interested in, $\mathbf{y} \in \mathbb{R}^m$ is the signal we actually can measure, and $\mathbf{\Phi} \in \mathbb{R}^{m \times n}$ is the sensing matrix, such that $m < n$.

Solving the sparse reconstruction problem relies on the assumption that the signal is sufficiently compressible in some basis or dictionary $\mathbf{\Lambda} \in \mathbb{R}^{n \times l}$. This implies that $\boldsymbol{\alpha} = \mathbf{\Lambda}\mathbf{s}$, with most coefficients of $\mathbf{s} \in \mathbb{R}^l$ being zero or close to zero. This dictionary is often learned from a training set representative of the images of interest[2] [4, 202]. We can then recover $\boldsymbol{\alpha}$ under certain conditions by solving the following minimization problem [94]:

$$\min_{\mathbf{s}} \|\mathbf{s}\|_1 \;\text{ subject to }\; \|\mathbf{y} - \mathbf{\Phi}\mathbf{\Lambda}\mathbf{s}\|_2^2 \leq \epsilon \tag{2.1}$$

where ϵ represents uncertainties in the measurements, such as sensor noise. This minimization is solved in a patch-based manner, that is the image is divided into a series of overlapping patches and each patch is reconstructed individually using Equation 2.1. All the reconstructed patches are subsequently merged, for example by computing a per-pixel average, to yield the final result.

A drawback of dictionary-based sparse coding approaches is that important spatial structures of the signal of interest can be lost due to the subdivision into mutually-independent patches.

[2] Alternatively, well-explored sparsity bases, such as the DCT or wavelets, could be used.

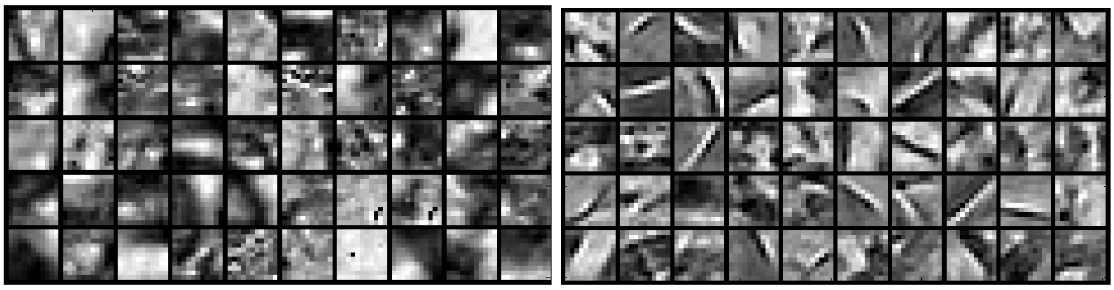


Figure 2.2: Left: sample atoms of learned dictionary trained on HDR images (patches are tonemapped for display). Right: sample filters learned with a convolutional sparse coding framework. The convolutional filter bank shows less redundancy, crisper features, and a larger range of feature orientations.

Further, patches (atoms) of the dictionaries learned with this approach are often redundant and contain shifted versions of the same features. This can be seen in Figure 2.2 (left), which shows sample atoms of a dictionary learned from HDR images. Moreover, as we show in Section 2.4.2 and Figure 2.5, due to the nature of the mathematical formulation (a linear combination of learned patches), these patch-based approaches can fail to adequately represent high-frequency, high-contrast image features, which are particularly important in HDR images.

An alternative to patch-based approaches is CSC, which instead is based on an image decomposition into spatially-invariant convolutional features, as explained in the following. Compared to the atoms of a dictionary, the learned filters of our CSC scheme (Figure 2.2 (right)) show a much richer variance (e.g., they span a larger range of orientations), which leads to better reconstructions.

Convolutional sparse coding models the signal of interest $\boldsymbol{\alpha} \in \mathbb{R}^n$ as a sum of sparsely-distributed convolutional features [137], that is $\boldsymbol{\alpha}$ is modeled as:

$$\boldsymbol{\alpha} = \sum_{k=1}^{K} \mathbf{d}_k * \mathbf{z}_k, \tag{2.2}$$

In this case, the dictionary is a convolutional filter bank formed by filters $\mathbf{d}_k$ of fixed spatial support $\sqrt{p} \times \sqrt{p}$, while $\mathbf{z}_k$ are sparse feature maps of size $\sqrt{n} \times \sqrt{n}$.

Consequently, the signal recovery can be performed by solving

$$\begin{aligned} &\underset{\mathbf{d},\mathbf{z}}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{x} - \sum_{k=1}^{K} \mathbf{d}_k * \mathbf{z}_k\|_2^2 + \beta \sum_{k=1}^{K} \|\mathbf{z}_k\|_1 \\ &\text{subject to} \quad \|\mathbf{d}_k\|_2^2 \leq 1 \quad \forall k \in \{1, \dots, K\}. \end{aligned} \tag{2.3}$$

Heide and colleagues [137] generalized this formulation to be able to handle incomplete data, as modeled by the general linear operator $\mathbf{M}$:

$$\begin{aligned} &\underset{\mathbf{d},\mathbf{z}}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{x} - \mathbf{M} \sum_{k=1}^{K} \mathbf{d}_k * \mathbf{z}_k\|_2^2 + \beta \sum_{k=1}^{K} \|\mathbf{z}_k\|_1 \\ &\text{subject to} \quad \|\mathbf{d}_k\|_2^2 \leq 1 \quad \forall k \in \{1, \dots, K\}. \end{aligned} \tag{2.4}$$

They also proposed a technique for efficiently solving this problem via splitting of the objective function.

2.3.2 *HDR image formation model*

Based on the film reciprocity equation [78], we can describe the image formation model at the sensor as:

$$\mathbf{y} = f(\mathbf{p} * \Delta t \mathbf{L}) \tag{2.5}$$

where $\mathbf{y} \in \mathbb{R}^n$ is the vectorized image captured at the sensor, Δt is the exposure time, $\mathbf{L} \in \mathbb{R}^n$ represents radiance values, and the function f models the camera response. The convolution by $\mathbf{p}$ is modeling the effect of the point spread function (PSF) of the optical system, which can also be expressed as a multiplication by a convolution matrix $\mathbf{P}$. Note that we use radiance $\mathbf{L}$ instead of irradiance since almost all modern cameras provide a nearly constant mapping between both magnitudes, compensating for angular effects [78, 169]. We optically modulate the light arriving at each pixel by placing a coded transmissivity mask $\mathbf{\Omega}$ on the sensor or by applying a spatially-coded exposure readout. This can be formulated as

$$\mathbf{y} = f(\mathbf{\Omega}\mathbf{P}\Delta t \mathbf{L}) \tag{2.6}$$

where $\mathbf{\Omega} \in \mathbb{R}^{n \times n}$ is a diagonal matrix containing the modulation code of the mask. For RAW images, we can assume a linear response of the digital sensor with respect to irradiance for all non-saturated pixels [185]. Thus, we can rewrite Equation 2.6 as $y = \zeta \mathbf{\Omega}\mathbf{P}\mathbf{L}$, where ζ is a scale factor modeling the linear response of the sensor and the influence of exposure time Δt. This scaling factor (and thus absolute radiance values) could be recovered by imaging a calibrated light source and scaling all radiance values accordingly. In our context, we aim at obtaining relative radiance values, therefore we can remove ζ and rewrite Equation 2.7 in normalized form as:

$$\mathbf{y} = \mathbf{\Omega}\mathbf{P}\mathbf{L}^* \tag{2.7}$$

where $\mathbf{L}^*$ represents relative radiance values. The mask $\mathbf{\Omega}$ will ensure that pixels are sampled with effectively different exposure values, so that in all image regions at least some of the pixels properly sample the dynamic range. The sparse reconstruction step described next will be in charge of obtaining the radiance values from these differently sampled pixels.

2.3.3 *Convolutional sparse HDRI coding*

Equation 2.4 allows for the recovery of contrast-normalized images in which part of the data is missing or unreliable, as given by matrix $\mathbf{M}$. In the case of HDR reconstruction, however, our captured image $\mathbf{y}$—as given by Equation 2.7—does not only have missing or unreliable data, but also differently exposed pixels due to matrix $\mathbf{\Omega}$. In the case of HDR imaging the unreliable data $\mathbf{M}$ corresponds to both saturated and noisy pixels. Incorporating the varying exposures $\mathbf{\Omega}$ we pose the convolutional reconstruction of radiance values as:

$$\underset{\mathbf{z}}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{y} - \mathbf{\Omega}\mathbf{M}\mathbf{P}\sum_{k=1}^{K} \mathbf{d}_k * \mathbf{z}_k\|_2^2 + \beta \sum_{k=1}^{K} \|\mathbf{z}_k\|_1 \tag{2.8}$$

where β controls the relative weight of the sparsity term. Note that, in contrast to Eqs. 2.3 and 2.4, we optimize only for $\mathbf{z}$, since we assume that we have already learned a dictionary of filters $\mathbf{d}$.

The dictionary of filters $\mathbf{d}$ is learned using Equation 2.4, and some of the learned filters are shown in Figure 2.2 (right). We learn the filters from a set of LDR images, after performing a local contrast normalization on these images. This amounts to learning from whitened data (normalized sigma and mean). As a consequence of this normalization, the formulation cannot be used directly in a generative model: While the correct scaling for recovery can be obtained during the optimization by finding the correct values in the sparse maps $\mathbf{z}$, the offset cannot. To solve this, we introduce an offset term $\mathbf{o}$ that we jointly estimate with the sparse feature maps. The smoothness is ensured by a quadratic smoothness constraint, leading to:

$$\underset{\mathbf{z}}{\operatorname{argmin}}\ \frac{1}{2}\|\mathbf{y} - \mathbf{\Omega M P}(\sum_{k=1}^{K} \mathbf{d}_k * \mathbf{z}_k + \mathbf{o})\|_2^2 + \beta \sum_{k=1}^{K} \|\mathbf{z}_k\|_1 + \lambda_s \|\nabla \mathbf{o}\|_2^2 \tag{2.9}$$

Thanks to this normalization, the filters generalize to different means and scales—which are obtained during the optimization—, and they are independent of dynamic range. We additionally observe that the learned filters have fewer data-specific features and are more general this way, and the learning converges in fewer iterations. Specific implementation details on the filter dictionary learning are given in Section 2.5.

We can elegantly fit this additional offset in the proposed optimization framework by expressing it as the convolution $\mathbf{o} = \mathbf{d}_{K+1} * \mathbf{z}_{K+1}$, where $\mathbf{d}_{K+1}$ is a Dirac delta, and Equation 2.9 thus becomes:

$$\underset{\mathbf{z}}{\operatorname{argmin}}\ \frac{1}{2}\|\mathbf{y} - \mathbf{\Omega M} \sum_{k=1}^{K+1} \mathbf{P d}_k * \mathbf{z}_k\|_2^2 + \beta \sum_{k=1}^{K} \|\mathbf{z}_k\|_1 + \lambda_s \|\nabla \mathbf{z}_{K+1}\|_2^2 \tag{2.10}$$

where λ_s controls the relative weight of the smoothness term. Note that only smoothness, and not sparsity, is enforced for this $\mathbf{z}_{K+1}$.

Finally, if we rewrite Equation 2.10 by substituting $\hat{\mathbf{M}} = \mathbf{\Omega M}$ and $\hat{\mathbf{d}}_k = \mathbf{P d}_k$, our problem can be written as the CSC problem shown in Equation 2.4, with the exception of the quadratic smoothness term:

$$\underset{\mathbf{z}}{\operatorname{argmin}}\ \frac{1}{2}\|\mathbf{y} - \hat{\mathbf{M}} \sum_{k=1}^{K+1} \hat{\mathbf{d}}_k * \mathbf{z}_k\|_2^2 + \beta \sum_{k=1}^{K} \|\mathbf{z}_k\|_1 + \lambda_s \|\nabla \mathbf{z}_{K+1}\|_2^2 \tag{2.11}$$

We solve this problem using a modification of the ADMM algorithm [36]. To do so, we need to reformulate Equation 2.11 to express the first two terms as a sum of functions, in the following form:

$$\underset{\mathbf{z}}{\operatorname{argmin}}\ \sum_{i=1}^{I} f_i(\mathbf{K}_i \mathbf{z}) + \lambda_s \|\nabla \mathbf{z}_{K+1}\|_2^2 \tag{2.12}$$

For more details on this transformation please refer to [137, Sec. 2.1 and 2.2]. Once this is done, the modified ADMM algorithm to solve for $\mathbf{z}$ in our case is shown in Algorithm 1. The update in line 2 of the algorithm is solved in the spectral domain, and thus the additional smooth constraint does not increase the computational cost significantly w.r.t. the original formulation [137]. Also, the filter size does not matter in our case, since we are performing the filter inversion in the frequency domain. This would not be computationally efficient with traditional CSC methods such as that of Szlam et al. [313]. Finally, $\mathbf{prox}_\phi$ refers to the proximal operator of a function ϕ as described in Parikh and Boyd's work [246].

Algorithm 1 ADMM for HDR recovery

```
1: for $k = 1$ to $V$ do
2:     $\mathbf{y}^{k+1} = \underset{\mathbf{y}}{\operatorname{argmin}}\ \|\mathbf{K}\mathbf{y} - \mathbf{z} + \boldsymbol{\lambda}^k\|_2^2 + \lambda_s \|\nabla \mathbf{z}_{K+1}\|_2^2$
3:     $\mathbf{z}_i^{k+1} = \mathbf{prox}_{\frac{f_i}{\rho}}(\mathbf{K}_i \mathbf{y}_i^{k+1} + \boldsymbol{\lambda}_i^k)$
4:     $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + (\mathbf{K}\mathbf{y}^{k+1} - \mathbf{z}^{k+1})$
5: end for
```

2.4 Analyzing convolutional sparse HDRI coding

In this section, we provide an analysis of the proposed framework, including choice of coded exposure patterns and algorithmic parameters. We also show advantages of this formulation over traditional, patch-based sparse reconstruction for HDR capture.

2.4.1 *Design of coded exposure patterns*

There are several factors to take into account when designing the optical mask $\mathbf{\Omega}$. First, it needs to have a high light throughput, to avoid noise and reduce required exposure time; second, its per-pixel transmissivity values e_i should cover a wide range of exposures (that is, e_{max}/e_{min} should be large); and third, it should facilitate practical implementation. We tested several configurations for the mask over a set of seven different images; in particular, these configurations were: binary, Gaussian, uniform, uniform with four fixed exposures, fixed pattern with four exposures, and interleaved exposure. In the following we detail the formulation for each mask, the motivation behind its testing, and its performance.

We initially tested and compared the performance three optical masks: a binary mask, a mask where exposure values are drawn from a Gaussian distribution ($\mathbf{\Omega}_G = \{e_i\ ;\ \ e_i \sim \mathcal{N}(0.6, 0.1)\}$), and a mask obtained by drawing values from a uniform distribution ($\mathbf{\Omega}_U = \{e_i\ ;\ \ e_i \sim \mathcal{U}(0, 1)\}$). The reconstruction results are shown in Figure 2.3. The binary mask is limited when modulating the incoming light, and, as a result, is very limited in terms of the recovered dynamic range; large saturated areas, for instance, will be impossible to recover since all the pixels will be degraded due to the binary sampling. Both the uniform and the Gaussian masks yield good results, and choosing between them represents a trade-off between transmissivity and dynamic range. The Gaussian mask offers better light throughput, but a more limited recoverable dynamic range: most of the values of the Gaussian distribution will be close to the mean, with few very low values. As a result, large bright areas (such as in Figure 2.3, around the sun) may still remain saturated. A uniform mask allows recovery of a larger dynamic range because it more uniformly samples the range of exposures, minimizing the risk of large under- or over-exposed areas even in scenes of very high dynamic ranges.

While a uniform mask works well in practice, for a practical hardware implementation having a low number of discrete exposure values is beneficial. We therefore compare the uniform mask $\mathbf{\Omega}_U$ with a uniform 4-exposure mask $\mathbf{\Omega}_F$, that is one in which each pixel randomly takes one of four exposure values $\{e_1..e_4\}$. We choose the exposure values such that the ratio e_{max}/e_{min} covers 6 f-stops, i.e., $e_4/e_1 = 2^6$; this, with the dynamic range of 1000:1 that a standard CMOS sensor has [93], allows us to recover up to 16 stops in dynamic range. Figure 2.4 shows the quality of the resulting reconstruction for $\mathbf{\Omega}_U$ and $\mathbf{\Omega}_F$, which can be seen to be very similar in both. Thus, $\mathbf{\Omega}_F$ allows us to recover a very similar range to the uniform one, without artifacts, and has an easier implementation. Consequently, in the remaining of this work, we opt for a uniform, 4-exposure pattern ($\mathbf{\Omega} = \mathbf{\Omega}_F$), since it offers the best trade-off between quality of the results—in terms of recovered dynamic range and absence of artifacts—, and ease of implementation in hardware. The exception to this is our hardware prototype (Section 2.5.1): since it exhibits significant light loss (mainly due to the LCoS and the beamsplitter) we do use a Gaussian mask to minimize the impact of the reduced light throughput. However, future chip designs with built-in per-pixel exposure will overcome this prototype's limitations; taking this into account the best option among the configurations we tested is $\mathbf{\Omega}_F$.

Additionally, to highlight the versatility of our reconstruction framework, we tested two additional exposure patterns which have been used before in the context of HDR imaging. Their results are also shown in Figure 2.4 (in the two rightmost images). In particular, we show a reconstruction result for a *fixed pattern* $\mathbf{\Omega}_P$, using four exposures (that is, the mask shows a repeating, fixed 2×2 pattern), and a result for an interleaved exposure pattern. The former has been proposed before for HDR imaging, but with the reconstruction done by means of interpolation [229], which

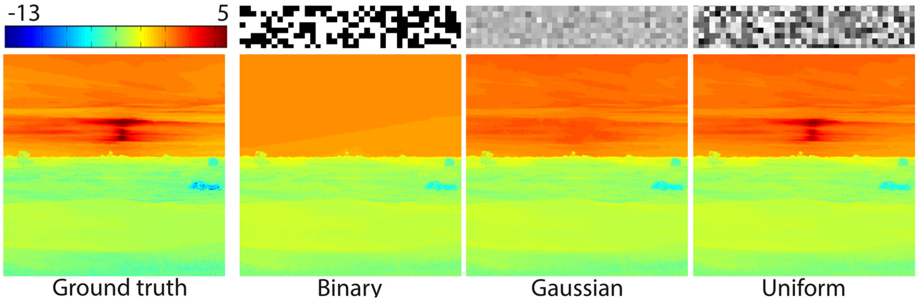


Figure 2.3: HDR images in false color (color scale shows f-stops) showing (from left to right): ground truth radiance, radiance recovered using a binary mask $\mathbf{\Omega}_B$ as optical code, a Gaussian mask $\mathbf{\Omega}_G$, and a uniform mask $\mathbf{\Omega}_U$ (more details in the text). The first two masks clearly fall short when recovering dynamic range, while the uniform one offers results very close to the original. The tonemapped ground truth image can be seen in Figure 2.4, left.

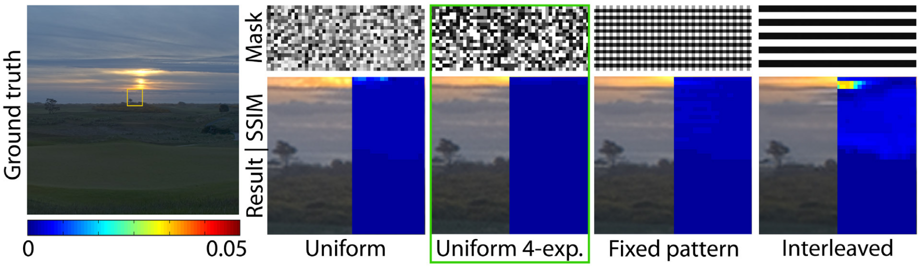


Figure 2.4: Left: a tonemapped HDR ground truth image. Right: quality of different optical masks when attempting to recover the ground truth scene radiance. From left to right: a uniform mask $\mathbf{\Omega}_U$, a 4-exposure mask $\mathbf{\Omega}_F$, an interleaved mask $\mathbf{\Omega}_I$, and a fixed pattern mask $\mathbf{\Omega}_P$. For each one, the left part shows the reconstructed image, and the right part the error with respect to the ground truth displayed as $(1 - SSIM)$ [336]. The top row shows a sample region of the corresponding mask. We choose $\mathbf{\Omega}_F$ for its ability to faithfully recover a wide dynamic range and its ease of implementation. Please refer to text for more details.

can lead to aliasing effects. The latter is inspired by the *Magic Lantern* software package, which offers a firmware upgrade to capture an interleaved exposure consisting of alternating rows with two different exposures $\mathbf{\Omega}_I$ for some off-the-shelf cameras. Our framework allows for a plausible result even with these exposure patterns.

2.4.2 *Advantage of CSC HDRI over patch-based approaches*

Patch-based sparse reconstruction approaches have been widely used in computational imaging problems [192, 195, 214]. In this section, we illustrate and explain how directly applying such approaches to the problem of HDR reconstruction from a single, exposure-coded image would produce undesired results in a number of cases.

We have already seen how the filters in our framework show a richer variance (less redundancy and a larger range of orientations) compared to traditional atoms in a learned dictionary (Figure 2.2). Consequently, CSC dictionaries are more descriptive and better capture the essence of the signals (e.g., they avoid the need to have shifted versions of the patches), which results in better reconstructions. More importantly, in patch-based approaches, the signal (a given image patch that is to be reconstructed) is represented as a linear combination of dictionary patches with their associated coefficients. This is problematic when attempting to reconstruct patches which contain very large contrast edges (common in HDR images), because an extremely large number of patches with high-valued coefficients is needed to properly reconstruct the edge. This is of course

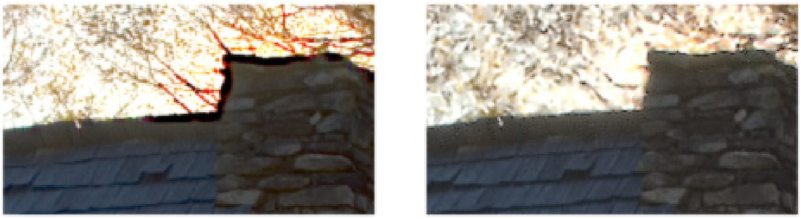


Figure 2.5: Detail of an HDR image reconstructed using a patch-based sparse reconstruction approach (left) and our convolutional sparse coding framework (right). The former is unable to recover very high-contrast sharp edges, while the latter offers good results in this case. The images are tonemapped for display using [209].

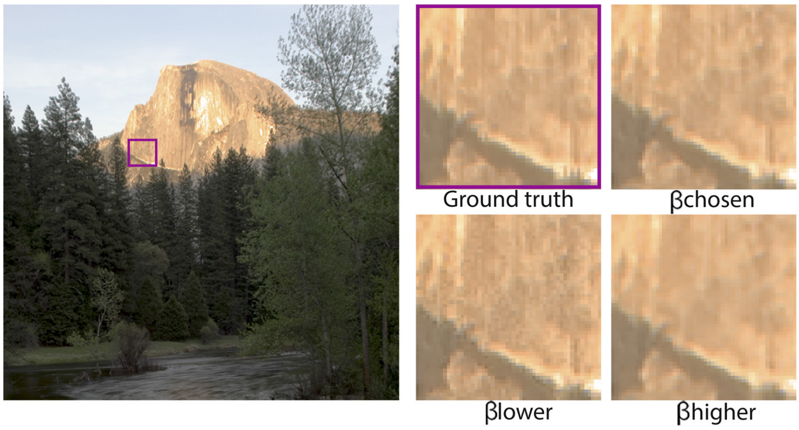


Figure 2.6: Reconstructed HDR image (tonemapped for display) showing the effect of *fi*, the relative weight of the sparsity term, in the optimization. Please refer to the text for details.

not only the case with learned dictionary patches, but also if any other basis (e.g., DCT) is used. As such, this problem was also encountered in the past in HDR image compression [210, Fig. 5]. Consequently, when reconstructing HDR images with a patch-based approach the reconstruction fails in the presence of very high contrast edges, yielding artifacts as shown in Figure 2.5, left. CSC, in contrast, can naturally handle these large contrast edges—as shown in Figure 2.5 (right)—thanks to the formulation of the signal as a sum of convolutions of the filters by sparse feature maps as opposed to a linear combination of dictionary elements.

Moreover, the convolutional sparse coding framework converges significantly faster than the patch-based approach (for which we use the well-known OMP algorithm [323]). Specifically, in an Intel Xeon E5-1620 @3.50GHz with 16GB RAM our CSC approach is around 2.5x faster.

2.4.3 *Optimization parameters*

As explained in Section 2.3.3, β controls the relative weight of the sparsity term with respect to the data term (see Equation 2.11). Increasing the value of β will therefore result in a degradation of the high frequencies in the reconstructed scene, since the feature maps $\mathbf{z}$ will be too sparse to represent fine details. Decreasing β, on the contrary, will lead to an excessive relative weight of the data term, which can result in artifacts due to approximations of non-linearities of the process (such as the quantization). Figure 2.6 shows this behavior. We choose an intermediate value of β, $\beta_{chosen} = 1.5 \cdot 10^{-5}$, which we use in all the reconstructions shown in this work.

The other relevant parameter in the optimization is the relative weight of the quadratic smoothness term, λ_s in Equation 2.11; we choose $\lambda_s = 0.5 \cdot 10^{-5}$. In this case, it is important that a good estimate of the offset term $\mathbf{z}_{K+1}$ is given as initial value to the optimization. We provide a blurred version of the captured LDR image divided by the optical mask, which yields good results and fast convergence.

2.5 Results

We show here reconstruction results using both existing HDR images[3], and data captured with our prototype camera. All results shown have been reconstructed using our single-shot method described in this work, with the same optical mask $\mathbf{\Omega}_F$ described in Section 2.4.1, consisting of four randomly sampled exposure values with $e_{max}/e_{min} = 2^6$, except where otherwise indicated. The filter bank $\mathbf{d}_k$ used for the reconstruction is learned from a collection of ten natural LDR images using the method proposed by Heide et al. [137]; a representative sample of these learned filters is shown in Figure 2.2 (right). When choosing the training images we learn the filters from, we found our framework robust enough to provide similar results when learned from different sets of images: Learning the filter bank from a dataset of images used in the work of Heide et al. or learning from tonemapped images from Fairchild's database (on a set not used for testing) yielded reconstructions which differed in less than 0.5 dB in PSNR. The size of the filters is determined by the resolution of the training data; the filters need to be large enough so they contain useful information, yet small enough not to overfit to specific features of the training data. We find that learning $K = 100$ filters of size 11×11 pixels fulfills these conditions for our data and works well for all the images tested. All HDR results shown have been tonemapped using the same algorithm [209]. We additionally compare our results to two other spatially varying exposure methods [128, 229]. For results using existing HDR images as input, we simulate the process of capturing the coded LDR image as follows: We first apply a convolution kernel $\mathbf{p}$ simulating the optical PSF of the camera, and modulate light arriving at the sensor multiplying the radiance values of the input HDR by our coded mask. We bracket these values taking into account that a typical CMOS sensor has a dynamic range of around $1000 : 1$. In doing so, we assume a reasonably well-exposed LDR image, but nevertheless we simulate the metering of a camera and take into account saturation and under-exposure by placing the sensor range so that the number of saturated and under-exposed pixels is minimized. Then we normalize these bracketed values and apply a camera response function[4]. Last, we quantify the resulting values to store the LDR image which will be used as input for the reconstruction.

Figure 2.7 shows four of our reconstructed HDR images. In addition to our reconstruction (top row), we show, for each scene, a false color image of the ground truth scene and our reconstruction (bottom row, split images) to show our ability to recover the large dynamic range present in the original scene. Since we recover relative radiance, and given the large dynamic range, we plot in false color $log2$ radiance normalized to the ground truth. Further, the insets in the top row show the error, computed as the square of the per-pixel difference between ground truth and our reconstruction, scaled for visualization purposes. We also report the PSNR for each one, which is always above 40 dB. This figure shows how our method is able to recover scenes with very high dynamic range, faithfully reproducing contrast in the original scene. More reconstructed scenes can be found in Figure 2.8, in which we show our reconstructed HDR image (top row), normalized luminance of sample scanlines, both recovered and ground truth (middle row), and two exposures of the reconstructed scene to better show the quality of the reconstruction across the dynamic range, including the challenging case of high-contrast sharp edges (bottom row).

Different from other common spatially varying exposure methods, our approach does not rely on interpolation of the captured samples to reconstruct the image. Instead, it exploits information of the structure of natural images through the learned convolutional filter bank, which greatly minimizes the presence of visible artifacts in areas of high contrast or very fine detail. We show this by explicitly comparing our results against the spatially varying exposure methods of Nayar

[3] We use images from the HDR Photographic Survey (`http://rit-mcsl.org/fairchild/HDR`), and the EMPA HDR Image Database (`http://www.empamedia.ethz.ch/hdrdatabase/index`).

[4] `http://www1.cs.columbia.edu/CAVE/software/softlib/dorf`

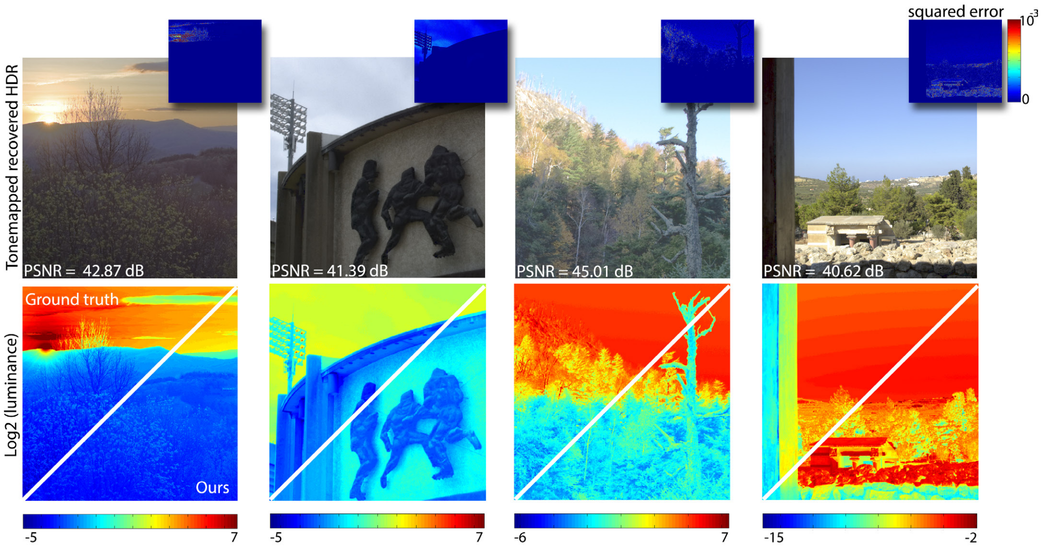


Figure 2.7: *Top row:* Recovered HDR images from a single-shot coded image (tone mapped using [209]), and PSNR values. The insets show the squared, per-pixel difference with respect to the ground truth luminance. *Bottom row:* False color (split) images depicting luminance of the original scene, and of our reconstructed scene; we use a base-2 logarithm to properly display the extremely large dynamic range.

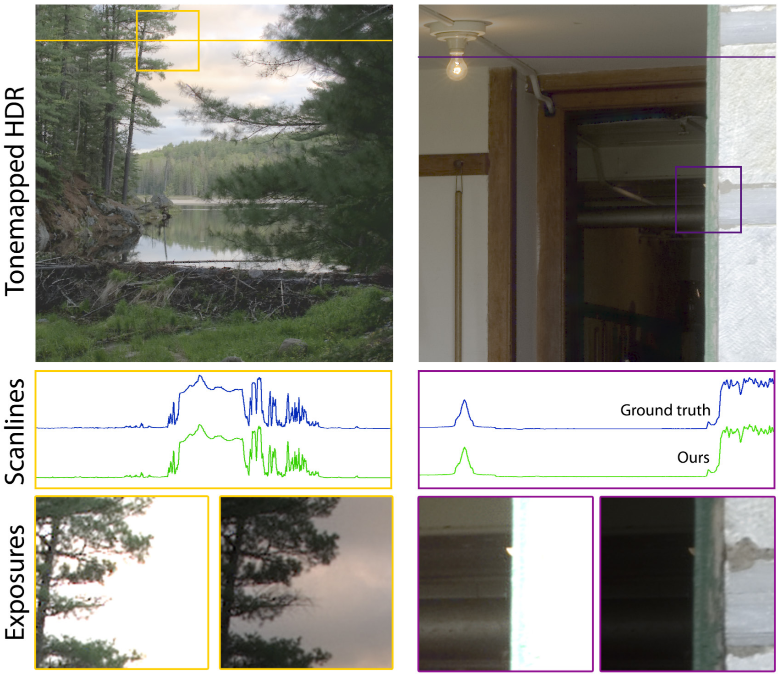


Figure 2.8: Additional results obtained by our technique for two HDR scenes. *Top row:* tonemapped HDR image (using [209]. *Middle row:* Normalized luminance plots for the corresponding marked scanlines for our recovered image (green curve) and the ground truth image (blue curve). *Bottom row:* Close-up of two exposures of the corresponding highlighted regions, displaying very high-contrast edges.

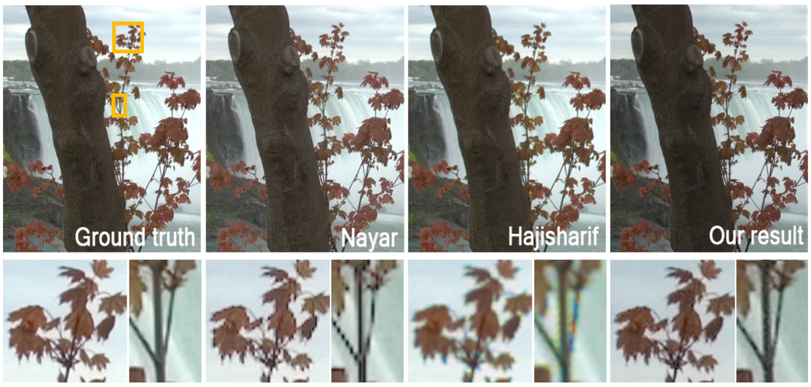


Figure 2.9: Comparison with two representative spatially varying exposure methods [128, 229]. The inherent interpolation step in such methods leads to visible artifacts in areas of high contrast or very fine detail.

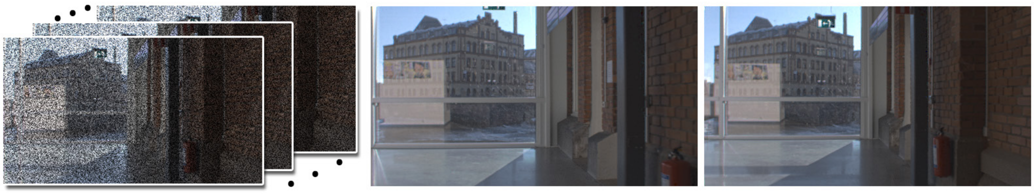


Figure 2.10: HDR reconstruction of an animated scene. *Left:* Coded sensor images using our optical code $\mathbf{\Omega}_F$. *Center and right:* Two example frames exhibiting temporal coherence in the reconstruction. Input video from the LiU HDR video repository.

et al. [229] and Hajisharif et al. [128], which makes use of the *Magic Lantern* software to capture interlaced, dual-ISO images. Our method preserves edges better, minimizing the aliasing artifacts that arise from the trade-off between spatial resolution and dynamic range in Nayar's method, while Hajisharif's method has difficulties recovering thin structures, such as the small branches of the tree (Figure 2.9).

Our technique can be applied to the reconstruction of HDR animated scenes as well, using the same optical code for each frame. Our reconstruction framework yields a very faithful recovery of the original signal, naturally leading to temporal coherence, without the need for explicit enforcement. We show this in Figure 2.10, using an existing HDR video from the LiU HDR video repository[5].The HDR video recovering is performed frame by frame from LDR capture simulations from the aforementioned HDR video.

Finally, our framework can also be used for compression of HDR images. Traditional techniques used for compression of images can fail when applied to HDR images, due to the high-contrast sharp edges that can be present in them. Consequently, techniques have been developed to compress this type of content [210]. Our framework allows for compression of HDR images, since we can represent them with a set of sparse feature maps. We have shown in Figure 2.5 how for HDR content we avoid artifacts that appear when codification and reconstruction with patch-based schemes is used. Note that DCT was also proven to not work well by Mantiuk et al. [210], requiring more complex processing for compression.

2.5.1 *Hardware prototype implementation*

Per-pixel exposure cameras are not commercially available yet, although a per-pixel exposure patent has already been filed by Sony Corporation [157]. We have built a prototype that simulates

[5] http://www.hdrv.org

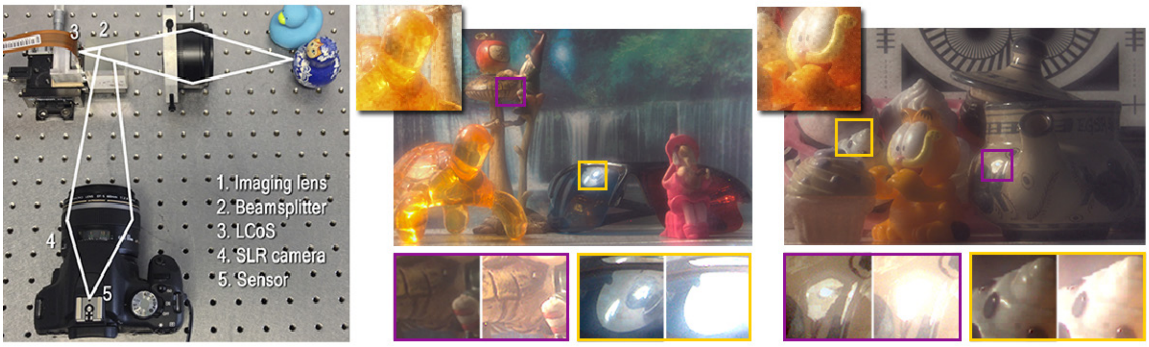


Figure 2.11: *Left:* Our prototype hardware implementation. Our optical system is made up of an imaging lens, a beamsplitter, an LCoS, and an SLR camera. Objects are placed for illustration purposes only; when photographing the scene, they are placed at a distance of 80 - 100 cm from the imaging lens. *Middle and right:* Two reconstructions of real scenes. For each scene we show the tonemapped HDR reconstruction (top), two different exposures of the highlighted areas revealing the dynamic range (bottom), as well as a partial detail of the LDR coded image captured at the sensor (inset).

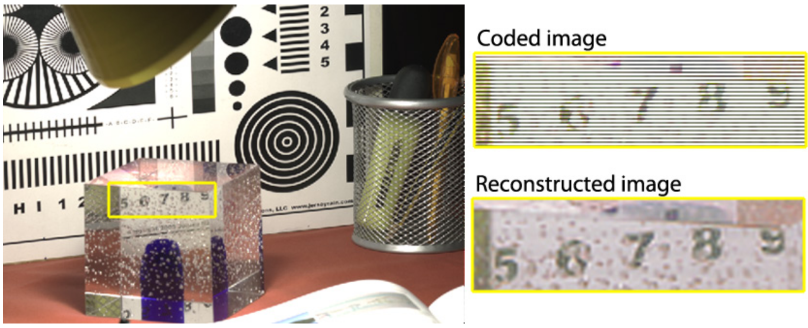


Figure 2.12: Reconstruction of an HDR image captured with dual ISO 100/800 with a Canon EOS 500D: original scene *(left)*, and close-ups of coded and reconstructed regions, the latter tonemapped using [209] *(right)*.

this feature to demonstrate our method with real scenes. To this end, we have implemented a capture system based on a liquid crystal on silicon (LCoS) display (Figure 2.11, left). This device, together with a beamsplitter and relay optics, simulates a Gaussian attenuation mask placed before the sensor. In this setup, the SLR camera lens (Canon EF-S 60 mm f/2.8 Macro USM) is focused on the LCoS, virtually placing the mask at the sensor. Our imaging lens is a Canon EF 50 mm f/1.8 II, focused at 50 cm; scenes are placed at 80 - 100 cm. The f-number of the system is f/2.8, the maximum of both lenses. Since a single pixel of the LCoS cannot be well-resolved with this setup, we treat LCoS pixels in blocks of 8 × 8 pixels, resulting in a mask with a resolution of 240 × 135. Figure 2.11 (right) shows results with real scenes captured with our prototype optical setup. The figure includes a close-up of the LDR coded image captured at the sensor, the final tone mapped HDR reconstruction, and several details with varying exposure levels. Our lab prototype is not artifact-free, although it demonstrates the viability of our approach. The LCoS displays some birefringence, decreased light throughput, and a severe loss of contrast, all of which degrade the LDR captured signal. Future chip designs such as the Sony patent could overcome these limitations. Nevertheless, our reconstruction does not introduce additional degradation in the results, as Figures 2.7 and 2.8 show.

Additionally, we have applied our technique to an image captured using an interlaced exposure with dual ISO 100/800 on a Canon EOS 500D camera with the *Magic Lantern* sofware. The result is shown in Figure 2.12.

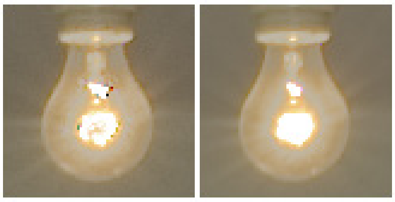


Figure 2.13: Example of limitations of our proposed HDR capture framework. This scene has a very large dynamic range (over 17 stops), since it captures both the very dark inside of the room and the bright light bulb outside. Therefore, if the inside is to be recovered, there is a saturated area in the captured image.

2.6 Discussion and conclusion

LIMITATIONS In some cases, it is possible that the image $\mathbf{y}$ captured with the optical mask contains large saturated areas despite the presence of the mask; the low transmissivity pixels of the mask typically prevent this, but in images with extremely large dynamic range it can happen. In these cases when no information at all is captured, the recovery may have some artifacts. An example of this is shown in Figure 2.13 with a light bulb. This light bulb is a close-up region of the scene in Figure 2.8 (right column). This scene has a very large dynamic range (over 17 stops), since it captures both the very dark inside of the room and the bright light bulb outside. Therefore, if the inside is to be recovered, there is a saturated area in the captured image $\mathbf{y}$. Nevertheless, as we show in this work, we are able to faithfully reconstruct scenes of very large dynamic range.

BENEFITS We have presented a framework for convolutional sparse coding of HDR images. From a single, optically coded image, we reconstruct dynamic range using a trained convolutional filter bank. Our approach follows a current trend in computational photography, leveraging the joint design of optical elements and processing algorithms. Once trained, the obtained filter bank can be used to reconstruct a wide variety of HDR images greatly differing from the training set. Since our reconstruction is based on a convolutional approach, it does not rely on the linear combination of patches common in sparse reconstruction methods; this greatly reduces reconstruction artifacts, in particular in high-contrast sharp edges present in HDR images. We are not limited to a restricted number of captured exposures, nor do we face the implicit trade-off between captured dynamic range and interpolation quality that other methods based on spatially-varying exposures face. In comparison to other CSC approaches, the algorithm we base our formulation on has demonstrated (see [137, Sec. 3]) that it has a lower complexity and better convergence than previously proposed methods for CSC [37, 38, 354], benefits which directly carry over to our method.

As an additional advantage, our framework naturally accounts for the optical PSF of the system, since we incorporate it in our model ($\mathbf{P}$ in Equation 2.10). Moreover, it can be easily extended to perform demosaicking, by properly designing matrix $\mathbf{M}$ in Equation 2.10, which models missing pixels. Last, we have not only built a physical prototype, but have also shown how our approach can yield good results with off-the-shelf consumer hardware that captures interleaved exposures using the Magic Lantern software.

FUTURE WORK The development of patents like Sony's per-pixel, double exposure method will progressively introduce varying exposure and optically modulated systems, thus allowing for increased capabilities of commercial cameras. Our optimization could incorporate explicit modeling of image noise to perform denoising in particularly noisy images. Finally, an exciting avenue of future work lies at the convergence between acquisition and display technologies, for the full plenoptic function and taking perceptual considerations into account [218]; compressive sensing and sparse coding techniques may be able to handle the high dimensionality of this challenging problem.

3. Convolutional sparse coding for high-speed video acquisition

ABOUT THIS CHAPTER

The work presented in this chapter has been published in three venues: some preliminary results were presented as a poster at the *International Conference on Computational Photography* (ICCP) 2015. Further results were presented at a national conference, the *Congreso Español de Informática Gráfica* (CEIG) 2015, where it was selected as one of the two best papers, and invited to submit an extended version to the journal *Computer Graphics Forum*, which was later published. While I led the line of work (under the supervision of Diego Gutierrez and Belen Masia), Elena Garces participated in the extension to *Computer Graphics Forum*, collaborating with the exploration of alternative reconstruction methods, and the generation of new results.

A. Serrano, D. Gutierrez, and B. Masia.
An In-Depth Analysis of Compressive Sensing for High Speed Video Acquisition.
In International Conference on Computational Photography (posters), 2015.

A. Serrano, D. Gutierrez, and B. Masia.
Compressive High-Speed Video Acquisition.
Proc. of the Spanish Conference on Computer Graphics (CEIG) 2015.

A. Serrano, E. Garces, D. Gutierrez, and B. Masia.
Convolutional Sparse Coding for capturing High Speed Video Content.
Computer Graphics Forum, Vol. 36 (8), 2017.

3.1 Introduction

During the last years, video capture technologies have seen large progress, due to the necessity of acquiring information at high temporal and spatial resolution. However, cameras still face a basic bandwidth limitation, which poses an intrinsic trade-off between the temporal and spatial dimensions. This trade-off is mainly determined by hardware restrictions, such as readout and analog-to-digital conversion times of the sensors. This makes capturing high speed video at high spatial resolutions *simultaneously* still an open problem.

Recent works try to overcome these limitations either with hardware-based approaches such as the camera array prototype proposed by Willburn et al. [342], or with software-based approaches, like the work of Gupta et al. [123], where the authors proposed combining low resolution videos with a few key frames at high resolution. Other approaches rely on computational imaging techniques, combining optical elements and processing algorithms [218, 339, 340]. For instance, a novel approach based on the emerging field of compressive sensing was presented [141, 195]. This technique allows to fully recover a signal even when sampled at rates lower than the Nyquist-Shannon theorem, provided that the signal is sufficiently sparse in a given domain. These works rely on this technique to selectively sample pixels at different time instants, thus coding the temporal information of a frame sequence in a single image. They then recover the full frame sequence from that coded image. The key assumption is that the time varying appearance of the captured scenes can be represented as a sparse linear combination of elements of an overcomplete basis (dictionary). This representation, and the subsequent reconstruction, is done in a patch-based manner, which is not free of limitations. Training and reconstruction usually takes a long time; moreover, the overcomplete basis needed for reconstruction has to be made up of atoms of similar nature to the video that is being reconstructed. This in turn imposes the need to use a specialized, expensive camera to capture such basis.

This work represents an extension over our previous work [288], where we i) performed an in-depth analysis of the main parameters defined in Liu's *patch-based* compressive sensing and sparse reconstruction framework [195]; ii) introduced the Lars/lasso algorithm for training and reconstruction, and showed how it improved the quality of the results; iii) presented a novel algorithm for choosing the training blocks, which further improved performance as well as reconstruction time; and iv), we further explored the existence of a good predictor of the quality of the reconstructed video. These contributions are now briefly summarized in Sections 3.4, 2.4, and 2.5, and in the Appendix A.

In this work, we additionally introduce a novel *convolutional sparse coding* (CSC) approach for high-speed video acquisition, and show how this outperforms existing patch-based sparse reconstruction techniques in several aspects. In particular, both training and reconstruction times are significantly reduced, while the convolutional nature of the atoms allows for reconstruction of videos using generic, content-agnostic dictionaries (see an example in Figure 3.1). This is due to the fact that the basis learned no longer needs to be able to reconstruct each signal block in isolation, instead allowing shiftable basis functions to discover a lower rank structure. In other words, image patches are no longer considered independent; interactions are modeled as convolutions, which translates into a more expressive basis which better reconstructs the underlying mechanics of the signal [38]. This completely removes the need to capture similar scenes using an expensive high-speed camera, and to explicitly train a dictionary, which significantly extends the applicability of our CSC framework. As an example, all the results shown in this work have been reconstructed using an existing dictionary containing *images of fruits* [137]. Note that we sample less than 15% of the pixels, which are additionally integrated over time to form a single image; despite this extremely suboptimal input data, we are able to successfully reconstruct high-speed videos of good quality. We provide source high speed videos, code, and results of our implementation in the project website[6].

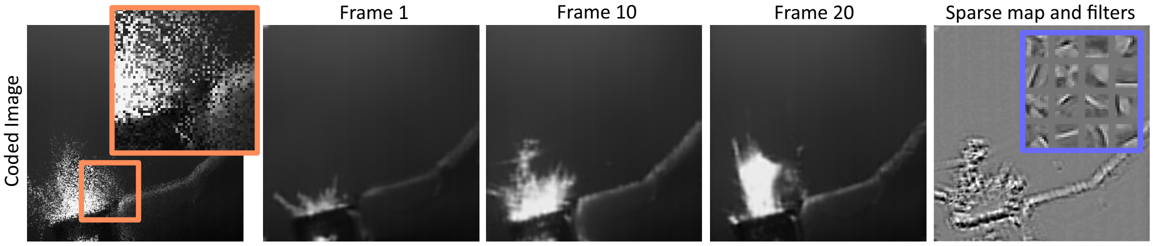


Figure 3.1: Reconstruction of a high speed video sequence from a single, temporally-coded image using convolutional sparse coding (CSC). The sequence shows a lighter igniting. *Left:* Coded image, from which 20 individual frames will be reconstructed; inset shows a close-up of the coded temporal information. *Middle:* Three frames of the reconstructed video. *Right:* CSC models the signal of interest as a convolution between sparse feature maps and trained filter banks: The image shows a sparse feature map for one of the frames, and the inset marked in blue some of the trained filters.

3.2 Related work

CODED EXPOSURES. Coded exposure techniques have been used to improve certain aspects of image and video acquisition in the field of computational photography. The goal is to optically code the incoming light before it reaches the sensor, either with coded apertures or shutter

[6] http://webdiis.unizar.es/~aserrano/projects/CSC-Video

functions. For instance, Raskar et al. [261] proposed the use of a *flutter shutter* to recover motion-blurred details in images. With the same purpose Gu et al. [120] propose the *coded rolling shutter* as an improvement over the conventional *rolling shutter*. Alternatively, codes in the spatial domain have been used for light field reconstruction [327], high-dynamic range imaging [229], to recover from defocus blur [216, 217], or to obtain depth information [190, 358].

COMPRESSIVE SENSING. The theory of compressive sensing has raised interest in the research community since its formalization in the seminal works of Candes et al. [47] and Donoho [86]. Numerous recent works have been devoted to applying this theory to several fields, including image and video acquisition. In one of the most significant works, the *Single Pixel Camera* of Wakin et al. [331], the authors introduce a camera prototype with only one pixel, which allows the reconstruction of complete images acquired with several captures under different exposure patterns. Other examples in imaging include the work of Marwah et al. [214], in which they achieve *light field* acquisition from a single coded image; high dynamic range imaging [276]; or capturing hyperspectral information [153, 192]. Recently, compressive sensing was proposed to reconstruct high-speed video from a single image [141, 195], combining coded exposure and dictionary learning. Some of the key design choices and parameters were analyzed in [288], leading to an improvement of the original design.

CONVOLUTIONAL SPARSE CODING. Convolutional sparse coding has quickly become one of the most powerful tools in machine learning, with many applications in signal processing, computer vision, or computational imaging, to name a few. Grosse et al. [118] introduced convolutional constraints to sparse coding, as well as an efficient minimization algorithm to make CSC practical, for the particular problem of 1D audio signals. Since then, many other works have extended this basic, canonical framework, with the goal of making it faster or more efficient (e.g. [37, 137, 172]). Bristow and Lucey [38] recently presented a large collection of examples covering different application domains, showing the fast spread and general applicability of CSC. Some examples of application domains include learning hierarchical image representations for applications in vision [57, 313]; decomposition of transient light transport [143]; imaging in scattering media [138]; or high dynamic range capture [290]. In this work we present a practical application of CSC for the particular case of high-speed video acquisition.

3.3 Background on sparse reconstruction

Compressive sensing has revolutionized the field of signal processing by providing a means to reconstruct signals that have been sampled at rates lower than what the Nyquist-Shannon theorem dictates. In general, this undersampling or incomplete acquisition of the signal is represented as:

$$\mathbf{y} = \mathbf{\Phi}\boldsymbol{\alpha} \tag{3.1}$$

where the signal of interest (in our case a video sequence) is represented by $\boldsymbol{\alpha} \in \mathbb{R}^m$, the captured signal (a coded image) is $\mathbf{y} \in \mathbb{R}^n$, with $n \ll m$, and $\mathbf{\Phi} \in \mathbb{R}^{n \times m}$ contains the sampling pattern and is called *measurement matrix*. An example of this sampling in the case of video acquisition is shown in Figure 3.2. Recovering the signal of interest having as input $\mathbf{\Phi}$ and $\mathbf{y}$ requires solving an underdetermined system of equations and finding a basis in which the signal of interest is sparse, and is known as the *sparse reconstruction* problem.

PATCH-BASED SPARSE CODING. To solve the sparse reconstruction problem posed by Equation 3.1, the signal of interest $\boldsymbol{\alpha}$ needs to be sparse, meaning that it can be represented in some alternative domain with only a few coefficients. This can be expressed as:

$$\boldsymbol{\alpha} = \sum_{i=1}^{N} \mathbf{\Lambda}_i \mathbf{s}_i \tag{3.2}$$

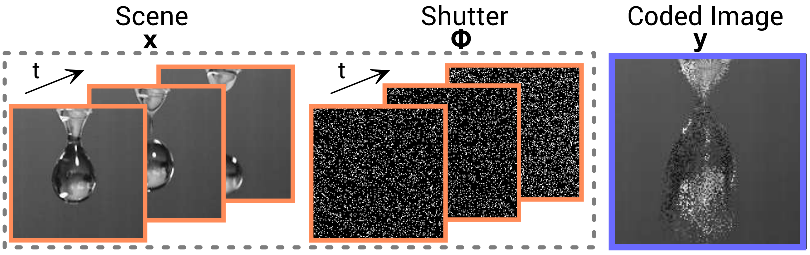


Figure 3.2: Video acquisition via compressive sensing. The signal of interest $\boldsymbol{\alpha}$ is sampled using a certain pattern $\mathbf{\Phi}$ which varies in time, and integrated to a single 2D image $\mathbf{y}$. Sparse reconstruction aims to recover $\boldsymbol{\alpha}$ from $\mathbf{y}$, knowing the sampling pattern $\mathbf{\Phi}$, a severely underdetermined problem. Note that the $\mathbf{\Phi}$ displayed in the image is an actual example of a pattern used for our captures.

where Λ_i are the elements of the basis that form the alternative domain, and $\mathbf{s}_i$ are the coefficients, which are in their majority zero or close to zero if the signal is sparse. Many natural signals, such are images or audio, can be considered sparse if represented in an adequate domain.

In order to reconstruct the original signal from the undersampled, acquired one, we jointly consider the sampling process (Equation 3.1) together with the representation in the sparse dictionary (Equation 3.2), yielding the following formulation:

$$\mathbf{y} = \mathbf{\Phi}\boldsymbol{\alpha} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{s} \tag{3.3}$$

where $\mathbf{\Lambda} \in \mathbb{R}^{m \times q}$ represents an overcomplete basis (also called *dictionary*) with q elements. If the original sequence $\boldsymbol{\alpha}$ is $s - sparse$ in the domain of the basis formed by the *measurement matrix* $\mathbf{\Phi}$ and the dictionary $\mathbf{\Lambda}$, it can be well represented by a linear combination of at most s coefficients in $\mathbf{s} \in \mathbb{R}^q$. Note that we are looking for a sparse solution; therefore, the search of the coefficients $\mathbf{s}$ has to be posed as a minimization problem. This optimization will search for the unknown $\mathbf{s}$ coefficients, seeking a sparse solution to Equation 3.3. This is typically formulated in terms of the L_1 norm, since L_2 does not provide sparsity and L_0 presents an ill-posed problem which is difficult to solve:

$$\min_{\mathbf{s}} \|\mathbf{s}\|_1 \ \textit{subject to} \ \|\mathbf{y} - \mathbf{\Phi}\mathbf{\Lambda}\mathbf{s}\|_2^2 \leq \epsilon \tag{3.4}$$

where ϵ is the residual error. Equation 3.4 is usually solved in a patch-based manner, dividing the signal spatially into a series of blocks (in the case of videos, the blocks are of size $p_x \times p_y \times p_z$), reconstructing them individually, and merging them to yield the final reconstructed signal.

Convolutional sparse coding. As an alternative, recent works propose modeling the $\boldsymbol{\alpha}$ as a sum of sparsely-distributed convolutional features [118]:

$$\boldsymbol{\alpha} = \sum_{k=1}^{K} \mathbf{d}_k * \mathbf{z}_k \tag{3.5}$$

where $\mathbf{d}_k$ are a set of convolutional filters that conform the dictionary, and $\mathbf{z}_k$ are sparse feature maps. The filters have a fixed spatial support, and the feature maps are of the size of the signal of interest.

Heide and colleagues [137] presented a formulation for the recovery of a signal $\boldsymbol{\alpha}$, modeled as in Equation 3.5, from a degraded signal $\mathbf{y}$ measured as shown in Equation 3.1:

$$\underset{\mathbf{z}}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\sum_{k=1}^{K} \mathbf{d}_k * \mathbf{z}_k\|_2^2 + \beta \sum_{k=1}^{K} \|\mathbf{z}_k\|_1 \tag{3.6}$$

In their paper, the authors also propose efficient algorithms to train the filter bank and to solve the minimization problem. Note that training the filter bank amounts to solving the minimization in Equation 3.6 optimizing for both the feature maps $\mathbf{z}_k$ and the filters $\mathbf{d}_k$.

CODED IMAGES FROM HIGH SPEED VIDEO SEQUENCES. In the case of video, the measurement matrix introduced in $\mathbf{\Phi}$ is implemented as a shutter function that samples different time instants for every pixel. The final image is thus formed as the integral of the light arriving to the sensor for all the temporal instants sampled with the shutter function:

$$I(x,y) = \sum_{t=1}^{T} H(x,y,t)A(x,y,t) \tag{3.7}$$

where $I(x,y)$ is the captured image, H the shutter function and A the original scene. In a conventional capture system $H(x,y,t) = 1 \ \forall \ x,y,t$ but in this case H should be such that it fulfills the mathematical properties of a measurement matrix suitable for sparse reconstruction, as well as the constraints imposed by the hardware. An easy way to fulfill the mathematical requirements is to build a random sampling matrix. However, since a fully-random sampling matrix cannot be implemented in current hardware, we use the shutter function proposed by Liu et al. [195], which can be easily implemented in a *DMD (Digital micromirror device)* or an *LCoS (Liquid Crystal on Silicon)* placed before the sensor, and approximates randomness while imposing additional restrictions to make a hardware implementation possible. In particular, the proposed shutter is implemented in a *LCoS*. Each pixel is only sampled once throughout the sequence, with a fixed bump length. We refer the reader to the work of Liu et al. [195] for more details about the coded shutter implementation.

In Section 3.4 we will show how to apply traditional, patch-based sparse coding to the problem of high speed video acquisition, and provide an analysis of the most important parameters. Section 3.5 will then offer an alternative solution, focused on efficiency, by modifying the convolutional sparse coding formulation in Equation 3.6. This alternative solution also lifts some of the restrictions imposed by the patch-based sparse coding approach, such as the need to build a dictionary whose atoms are similar to the videos that are going to be reconstructed.

3.4 Patch-based sparse coding approach

This section describes the specifics of how to solve the sparse reconstruction problem using a patch-based approach (described in Section 3.3) for high speed video. The mathematical formulation is given by Eqs. 3.2 to 3.4; here we describe how we train the dictionary $\mathbf{\Lambda}$, and how we solve the optimization problem in Equation 3.4. We summarize the main ideas here, and provide more details in Appendix A.

LEARNING HIGH SPEED VIDEO DICTIONARIES. We have captured a database of high-speed videos which we use for training and validation of our techniques. The database consists of 14 videos captured at captured at 1000 frames per second with a high-speed Photron SA2 camera, part of which are used for training, and part for testing. The Photron SA2 provides up to 4 Megapixels at a rate of 1000 fps. The acquisition setup is shown in Figure 3.3. Our database provides scenes of different nature, and a wide variety of spatial and temporal features. Representative frames of the videos in the database are shown in Appendix A. We learn the fundamental building blocks (atoms) from our captured videos, and create an overcomplete dictionary. For training, we use the DLMRI-Lab implementation [264] of the K-SVD [4] algorithm, which has been widely used in the compressive sensing literature. We propose an alternative to random selection that maximizes the presence of blocks with relevant information, by giving higher priority to blocks with high variance.

RECONSTRUCTING HIGH SPEED VIDEOS. Once the dictionary $\mathbf{\Lambda}$ is trained, and knowing the measurement matrix $\mathbf{\Phi}$, we need to solve Equation 3.4 to estimate the $\mathbf{s}$ coefficients and reconstruct the signal. Many algorithms have been developed for solving this minimization problem for compressive sensing reconstruction. We use the implementations available in the SPArse Modeling Software (SPAMS) [203].

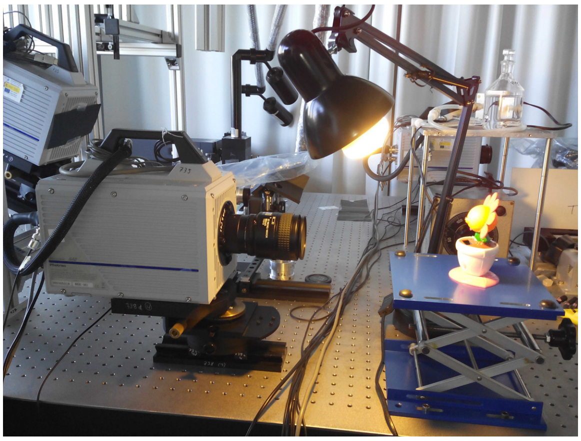


Figure 3.3: Setup for the acquisition of our high speed video database with a Photron SA2 camera. In order to capture videos with high quality the scene must be illuminated with a strong, direct light.

3.5 Introducing CSC for high-speed video acquisition

The use of *patch-based, conventional* sparse coding in the recovery of high-speed video produces good results, as we will show in Section 2.5. However, *convolutional* sparse coding (CSC) can offer significant improvements: First, learning convolutional filters allows for a richer representation of the signal, since they span a larger range of orientations and are spatially-invariant, as opposed to the patches learnt in a conventional dictionary. Second, due to their convolutional nature, dictionaries made up of filter banks are more versatile, in the sense that they are content-agnostic: they do not need to contain atoms of similar nature to the signals that are to be reconstructed with them. Finally, reconstruction time is a major bottleneck in patch-based approaches, but is significantly reduced in a convolutional framework; this is especially important when dealing with video content.

Recent efficient solutions for CSC have been proposed for images [137, 290]. In principle, adapting these solutions to video could be done simply by extending them to three dimensions (x-y-t), with 3D filters $\mathbf{d}_k$ and 3D feature maps $\mathbf{z}_k$. The optimization in Equation 3.6 could then be solved in a manner analogous to 2D [137, Algorithm 1]. Doing this, however, is tremendously computationally expensive[7]. We therefore revert to a 2D formulation instead, which is computationally manageable, and adapt it to be able to properly deal with video content, as explained next.

In our proposed formulation, the sparse feature maps $\mathbf{z}_k$ and the filters $\mathbf{d}_k$ remain two-dimensional; thus their convolution yields two-dimensional images, which correspond to the individual frames. This per-frame reconstruction of the video could be achieved using the optimization in Equation 3.6. To adapt this solution to video, we impose an additional constraint in the temporal dimension, which enforces sparsity of the first-order derivatives over time. The resulting reconstruction process becomes:

$$\underset{\mathbf{z}}{\text{argmin}} \; \frac{1}{2}\beta_d \|\mathbf{y} - \mathbf{\Phi} \sum_{k=1}^{K} \mathbf{d}_k * \mathbf{z}_k\|_2^2 + \beta_1 \sum_{k=1}^{K} \|\mathbf{z}_k\|_1 + \beta_2 \sum_{k=1}^{K} \|\nabla_t \mathbf{z}_k\|_1 \tag{3.8}$$

The operator ∇_t represents the first order backward finite difference along the temporal dimension: $\nabla_t \mathbf{z}_k = (\mathbf{z}_k^t - \mathbf{z}_k^{t-1}) \; \forall t \in \{2, \ldots, T\}$, where T is the number of frames being reconstructed. β_d, β_1 and β_2 are the relative weights of the data term, the sparsity term, and the temporal smoothness term, respectively.

[7] To give a practical example: 32GB of RAM allow training a maximum of *100* filters of size $11 \times 11 \times 10$; while this number of filters is adequate for 2D images, it is insufficient in the presence of the extra dimension in video.

A modified ADMM algorithm can be used to solve this problem by posing the objective as a sum of closed, convex functions f_j as follows:

$$\underset{\mathbf{z}}{\operatorname{argmin}} \sum_{j=1}^{J} f_j(\mathbf{K}_j\mathbf{z}) \tag{3.9}$$

where $J = 3$, $f_1(\xi) = \frac{1}{2}\beta_d\|\mathbf{y} - \mathbf{\Phi}\xi\|_2^2$, $f_2(\xi) = \beta_1\|\xi\|_1$, and $f_3(\xi) = \beta_2\|\xi\|_1$. Consequently, the matrices $\mathbf{K}_j$ are $\mathbf{K}_1 = \mathbf{D}$, $\mathbf{K}_2 = \mathbb{I}$, and $\mathbf{K}_3 = \nabla$, where $\mathbf{D}$ is formed by the convolution matrices corresponding to the filters $\mathbf{d}_k$, and ∇ is the matrix corresponding to the first order backward differences in the temporal dimension, as explained above. We use the implementation of ADMM (*Alternating Direction Method of Multipliers*) described in Algorithm 2 to solve Equation 3.9. More details about this implementation can be found in the literature [9, 137, Algorithm 1].

Algorithm 2

```
for $i = 1$ to $I$ do
    $\mathbf{y}^{i+1} = \underset{\mathbf{y}}{\operatorname{argmin}}\ \|\mathbf{K}\mathbf{y} - \mathbf{z} + \boldsymbol{\lambda}^i\|_2^2$
    $\mathbf{z}_j^{i+1} = \mathbf{prox}_{\frac{f_j}{\rho}}(\mathbf{K}_j\mathbf{y}_j^{i+1} + \boldsymbol{\lambda}_j^k)\ \forall j \in \{1, \dots, J\}$
    $\boldsymbol{\lambda}^{i+1} = \boldsymbol{\lambda}^i + (\mathbf{K}\mathbf{y}^{i+1} - \mathbf{z}^{i+1})$
end for
```

3.6 Analysis

In this section we discuss implementation details of the system, and we perform an analysis of the two approaches proposed in Section 3.4 and Section 3.5, exploring parameters of influence for both methods.

We now analyze the influence of the key parameters for both approaches: patch-based and convolutional sparse coding, and find the parameter combination yielding the best results. None of the test videos were used during training. As measures of quality, we use PSNR (Peak Signal to Noise Ratio), widely used in the signal processing literature, and the MS-SSIM metric [335], which takes into account visual perception. The complete analysis can be found in Appendix A.

3.6.1 *Patch-based sparse coding*

implementation details : We use the K-SVD algorithm [4] to train our dictionary and the LARS-Lasso solver [91] for solving the minimization problem. In order to achieve faithful reconstructions it is important that each atom (patch) is large enough to contain significant features (such as edges or corners), but not too large for avoiding learning very specific features of the training videos (and thus overfitting). We have tested several patch sizes (results included in Appendix A) and we have chosen the size yielding better quality in the results: $7pixels \times 7pixels \times 20frames$.

training a dictionary : The amount of blocks (3D patches) resulting from splitting the training videos is unmanageable for the training algorithm; thus the dimensionality of the training set has to be reduced. The straightforward solution is to randomly choose a manageable amount of blocks. However, a high percentage of these do not contain meaningful information about the scene (as in static backgrounds). We thus explore several other ways to select the blocks for training:

- **Variance sampling**: We calculate the variance for each block and bin them in three categories: high, medium and low variance. Then we randomly select the same amount of blocks for every bin to ensure the presence of high variance blocks in the resulting set.
- **Stratified gamma sampling**: We sort the blocks by increasing variance and sample them with a gamma curve ($f(x) = x^{\gamma}$). We analyze the effect of $\gamma = 0.7$, which yields a curve closer to a linear sampling, and $\gamma = 0.3$. The goal of this stratification is to ensure the presence of all the strata in the final distribution. We divide the range uniformly and calculate thresholds for the strata applying the gamma function. Then we randomly choose a sample from every strata and remove that sample from the original set. This process is iterated until the number of desired samples is reached.
- **Gamma sampling**: We choose directly samples from the original set following a gamma curve sampling. We also test values of $\gamma = 0.7$ and $\gamma = 0.3$.

Figure 3.4 shows results for one of the tested videos, with dictionaries built with the different selection methods explained above. Results for all the videos tested were consistent. *Random* and *Variance sampling* clearly outperform the other methods, with the *Variance sampling* yielding slightly better results.

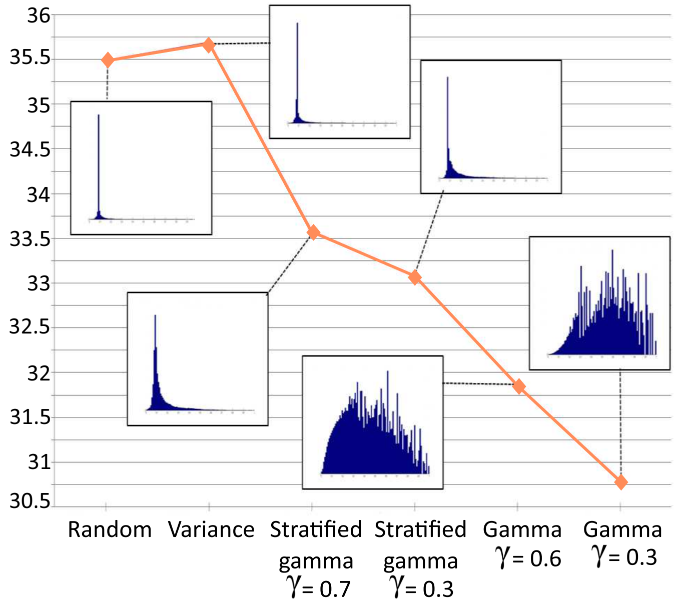


Figure 3.4: Quality of the reconstruction (in terms of PSNR) for a sample video (*PourSoda*) as a function of the method used to select training blocks for learning the dictionary. For each method we show an inset with the histogram of variances of video blocks of the resulting training set.

3.6.2 *Convolutional sparse coding*

IMPLEMENTATION DETAILS : We use a trained dictionary of 100 filters of size 11 × 11 pixels. We have performed tests with different filter sizes, and we have found the reconstruction quality to be very similar between different sizes (see Appendix A for results). The convolutional nature of the algorithm makes it more flexible, and unlike the patch-based approach, more robust towards variations in the filters size. Nevertheless, we chose to train the filters with a size of 11 × 11 pixels because they yield slightly better results. Regarding the amount of filters, we found that 100 filters are enough to make the algorithm converge in the training.

TRAINING A DICTIONARY : One of the theoretical advantages of convolutional sparse coding over the patch-based approach is that the learned dictionaries need not be of similar nature to the signal being reconstructed. To analyze this, we compare the quality of the reconstruction for two different dictionaries: A generic dictionary obtained from the *fruits* dataset (it simply contains pictures of fruits) provided by Heide et al. [137], and a specific dictionary trained with a set of frames from our captured video database (different from the ones we then reconstruct). Figure 3.5 shows that the quality of the reconstructions using both dictionaries is very similar for all the videos analyzed. This confirms the theory for the particular case of high-speed video reconstruction; as a consequence, we no longer need to acquire specific data to train dictionaries, adapted to every particular problem. We use this generic *fruit* dictionary for all our results.

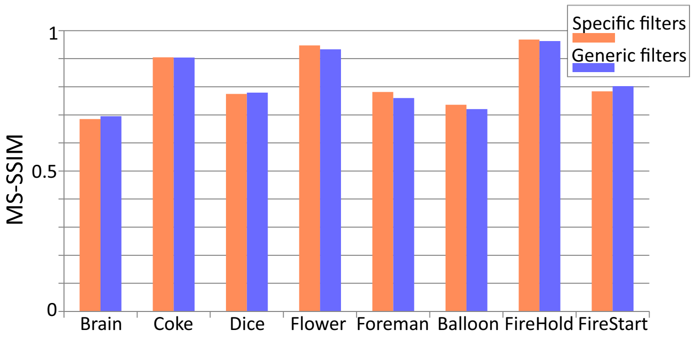


Figure 3.5: Quality of the reconstruction (in terms of the quality metric MS-SSIM) for two different dictionaries of filters: A specific one trained with frames from our video database (orange bars), and a generic one trained with the *fruits* dataset (blue bars). For all the videos analyzed (x-axis) the quality of the reconstruction is very similar with both dictionaries, showing our CSC-method does not require training a specific dictionary.

CHOOSING THE BEST PARAMETERS : We analyze the relative weight of each of the three terms in Equation 3.8 We set β_1, which weights the sparsity term, to 10, and vary parameter β_d, which weights the data term, and parameter β_2, which weights the temporal smoothness term. Based on the MS-SSIM results shown in Figure 3.6, we choose for all our reconstructions the values $\beta_d = 100$ and $\beta_2 = 1$.

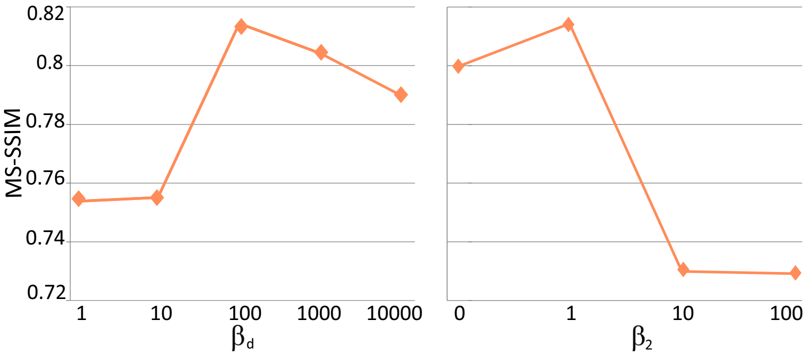


Figure 3.6: Analysis of parameters β_d and β_2 in Equation 3.8, which control the weights of the data term and the temporal smoothness term, respectively. We plot the mean MS-SSIM value from eight reconstructed videos.

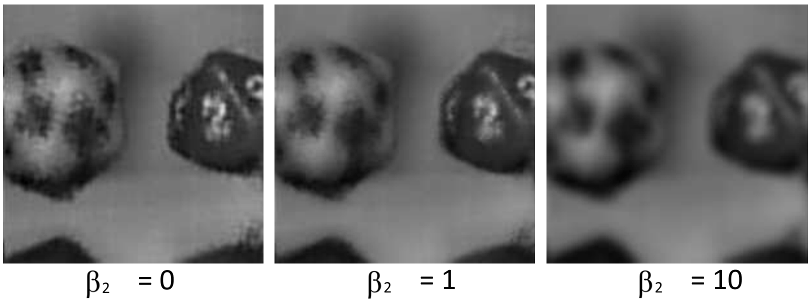


Figure 3.7: Effect of our proposed temporal smoothness term on the reconstructed videos. *From left to right:* Result of CSC without this term ($\beta_2 = 0$), and two increasing values for the weight of the temporal smoothness term in the optimization β_2. Not including this term yields results with many artifacts due to the low sampling rate of each image separately, while too high a value tends to over-blur the result.

3.7 Results and discussion

In this section we show and discuss our results with both sparse coding approaches: patch-based and convolutional. In recent work, Koller et al. [171] have performed an analysis of several state-of-the-art approaches for high spatio-temporal resolution video reconstruction with compressed sensing. They prove that the approach proposed by Liu et al. [141, 195] achieves better reconstruction qualities than other state-of-the-art approaches. Therefore we compare the results of our convolutional sparse coding approach with our implementation of the framework proposed by Liu et al.

The parameters used in all the reconstructions are derived from the analysis in the previous section. The videos used for training in the patch-based approach are different from the reconstructed ones (see Appendix A), whereas for CSC we use an existing, generic dictionary trained from images of fruits [137]. We have coded 20 frames in a single image. This number yields a good trade-off between quality and speed-up of the reconstructed video. For each frame, we sample less than 15% of the pixels. Despite this huge loss of information, we are able to reconstruct high-speed videos of good quality.

In general, both techniques yield results of similar quality, with a slight advantage for the patch-based approach in terms of MS-SSIM [335] values, of about 0.05 on average (see also Figure 3.8). However, as discussed earlier in the chapter, there are two important shortcomings of this approach: First, the need to train a dictionary made up of atoms containing similar structures to the reconstructed videos; otherwise the quality of the reconstruction degrades significantly. Second, training and reconstruction times are rather long (see Table 3.1 for reconstruction times). To overcome these, we introduced a second solution, based on convolutional sparse coding. This is not only significantly faster, but it also allows us to bypass the need to capture and train a dictionary, as discussed in Section 2.4.

As explained in this chapter, a naïve 3D CSC approach quickly becomes computationally intractable, due to the huge convolutional matrices involved. On the other hand, reverting to a 2D, per-frame solution yields many artifacts in the results due to the low sampling rate and the lack of temporal stability, as we show in Figure 3.7 (leftmost image) and Appendix A. We therefore adapted the 2D convolutional sparse coding approach to our problem, taking advantage of the coded temporal information by enforcing sparsity of the derivatives in time while solving the optimization. Figure 3.7 (middle and right images) shows the effect of this term in the optimization, while Figures 3.9 and 3.10 show additional results with each of the two techniques (patch-based and CSC-based). Please refer to project website for the full videos.

Last, Table 3.1 shows reconstruction times for all the videos shown in this work; on average, our CSC-based approach is 14x faster than a patch-based solution.

Table 3.1: Reconstruction times for eight videos (20 frames each) with convolutional sparse coding and with patch-based sparse coding. Note the great speed-up achieved by the former.

	Time (in seconds)	
	Convolutional	Patch-based
Brain	249	4,208
Coke	239	3,239
Dice	236	3,175
Flower	237	3,524
Foreman	237	3,746
Balloon	236	3,275
FireHold	237	3,071
FireStart	238	3,180

3.8 Conclusions

Computational imaging aims at enhancing imaging technology by means of the co-design of optical elements and algorithms; capturing and displaying the full, high-dimensional plenoptic function is an open, challenging problem, for which compressive sensing and sparse coding techniques are already providing many useful solutions. In this chapter we have focused on the particular case of high-speed video acquisition, and the intrinsic trade-off between temporal and spatial resolution imposed by bandwidth limitations. We have presented two sparse coding approaches, where we code the temporal information by sampling different time instants at every pixel. First, we have analyzed the key parameters in the patch-based sparse coding approach proposed by Liu et al. [141, 195], which have allowed us to offer insights that lead to better quality in the reconstructed videos. We then have introduced a novel convolutional sparse coding framework, customized to enforce sparsity on the first-order derivatives in the temporal domain. The convolutional nature of the filter banks used in the reconstruction allowed for a more flexible and efficient approach, compared with its patch-based counterpart. We bypass the need to capture a database of high-speed videos and train a dictionary, while reconstruction times improve significantly.

Many exciting venues for future research lie ahead. For instance, our strategy to impose an additional constraint in the temporal dimension is motivated by the fact that, due to the size of the convolutional matrices required to train the dictionary, it is not feasible to deal with (x-y-t) blocks directly. This is currently the main limitation of our approach, and it would be interesting to investigate other strategies in follow up work. Last, we hope that the development of computational techniques like ours will progressively allow the development of commercial imaging hardware with enhanced capabilities.

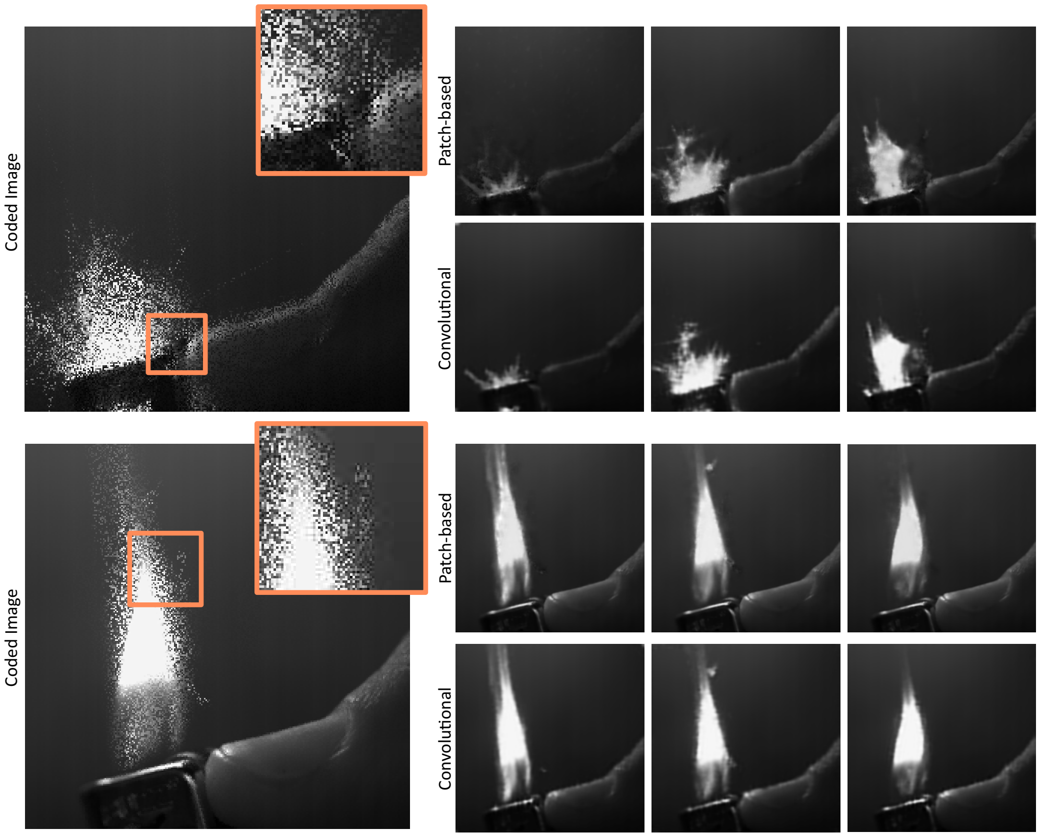


Figure 3.8: Representative frames of two reconstructed videos: *FireStart* (*top*) and *FireHold* (*bottom*). The sequences show a lighter at different stages of ignition. We show reconstruction results for the two approaches discussed in this work. *Left:* Input coded image, from which 20 frames will be reconstructed (inset shows a close-up). *Right:* Three sample frames of the reconstructed sequence, with the patch-based approach (*top row*) and with our CSC approach (*bottom row*).

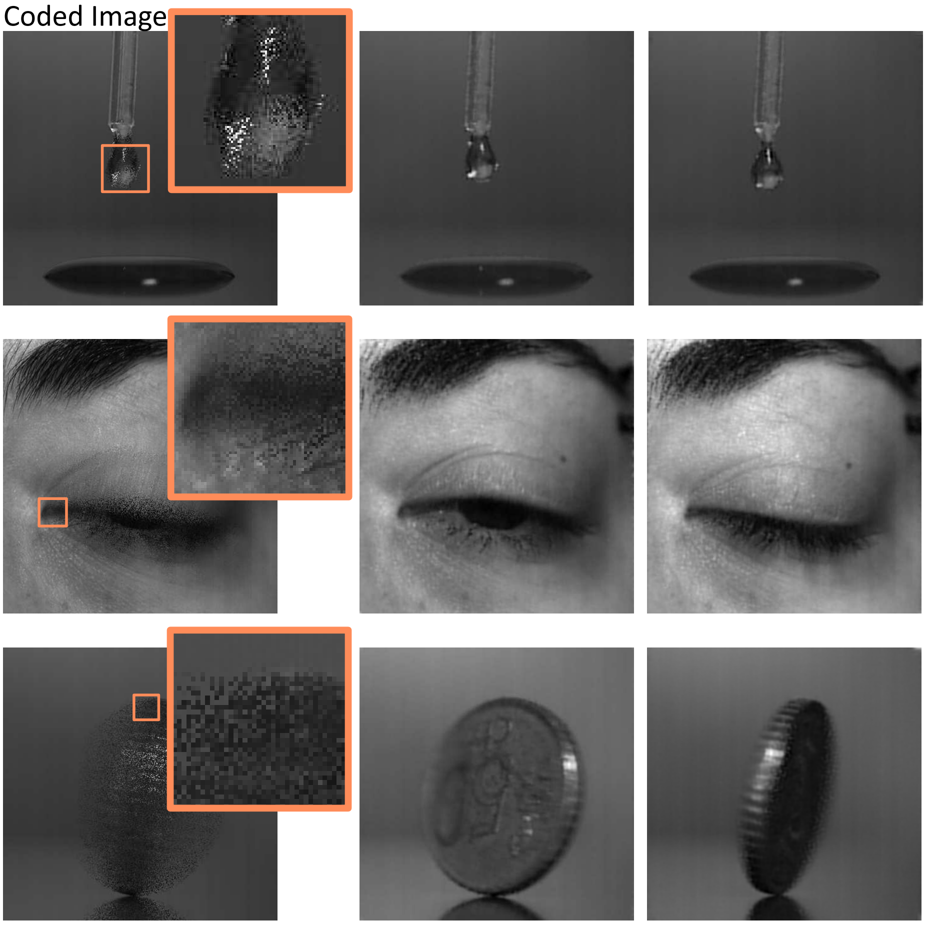


Figure 3.9: Additional video sequences reconstructed using our CSC-based approach. *Left:* coded image which serves as input to the reconstruction algorithm; inset shows a close-up. *Right:* Two of the frames reconstructed for each sequence. Detail is recovered despite the large loss of information undergone during sampling. Note that the blur in the coin of the bottom row is not motion blur but due to limited depth of field instead.

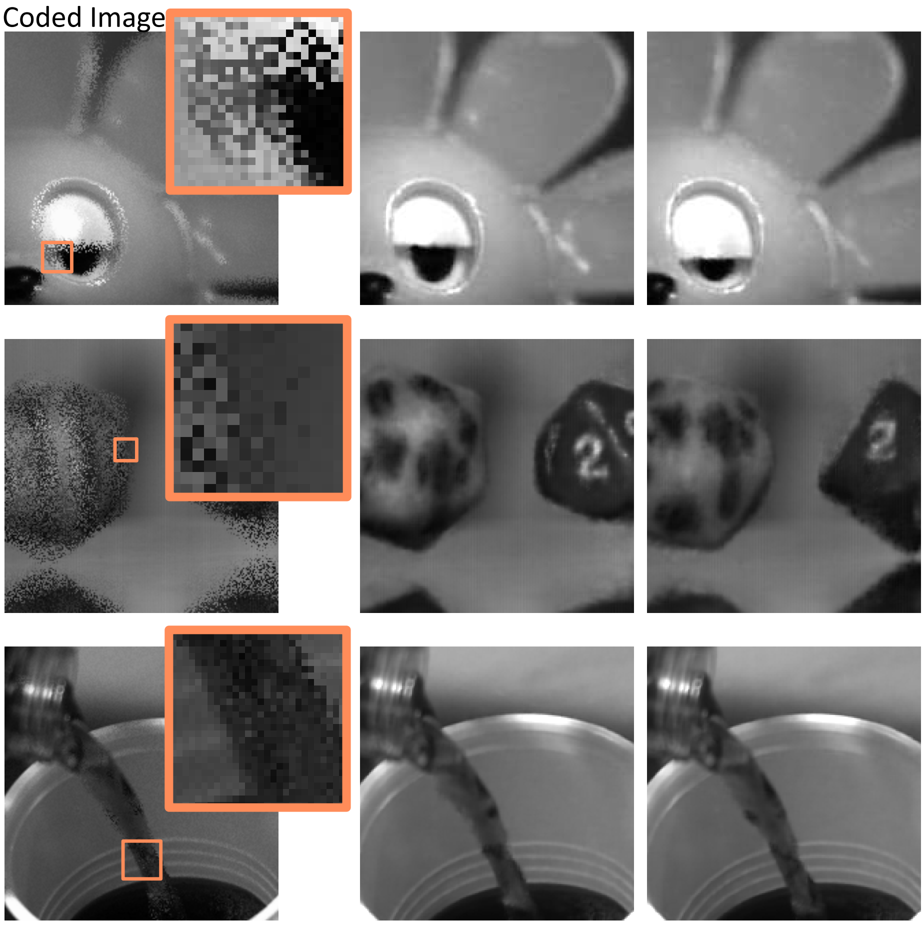


Figure 3.10: Additional video sequences reconstructed using patch-based sparse reconstruction. *Left:* coded image which serves as input to the reconstruction algorithm; inset shows a close-up. *Right:* Two of the frames reconstructed for each sequence.

Part III

Material appearance

In this part we build an intuitive, perceptually-based editing space for captured data by means of a large scale user study. We evaluate our space, and we show that is useful for several applications. We then focus in the particular application of gamut mapping, and propose a two-step optimization algorithm that takes into account perception in the mapping process. We evaluate our algorithm with objective and subjective metrics and demonstrate its performance.

4. An intuitive control space for material appearance

ABOUT THIS CHAPTER

The work presented in this chapter has been published in tree venues: preliminary results were presented as a poster at SIGGRAPH 2016. Further work was later presented at SIGGRAPH Asia 2016 and published in ACM Transactions on Graphics. In this work we propose a navigating space for measured materials based on perceptual attributes. This work was developed during my internship at Max-Planck Institute for Informatics. A follow-up (Section 4.9) of this work was carried out as the final degree project of Sandra Malpica, co-supervised by Belen Masia and myself, and it was presented at the national conference *Congreso Español de Informática Gráfica* (CEIG) 2017. Additionally, some highlights of this work were presented by Julio Marco in the *EGSR Material Appearance Modeling Workshop*.

A. Serrano, D. Gutierrez, K. Myszkowski, H.-P. Seidel, and B. Masia.
Intuitive editing of material appearance.
In ACM SIGGRAPH (posters), 2016.

A. Serrano, D. Gutierrez, K. Myszkowski, H.-P. Seidel, and B. Masia.
An intuitive control space for material appearance.
ACM Transactions on Graphics, Vol. 35 (6), 2016.

S. Malpica, M. Barrio, D. Gutierrez, A. Serrano, and Belen Masia.
Improved Intuitive Appearance Editing based on Soft PCA.
Proc. of the Spanish Conference on Computer Graphics (CEIG) 2017.

J. Marco, A. Serrano, A. Jarabo, B. Masia, and D. Gutierrez. Intuitive editing of visual
appearance from real-world datasets.
In EGSR Material Appearance Modeling Workshop, 2017.

4.1 Introduction

Measurement techniques for material appearance are gaining in accuracy, speed, efficiency, and ease of use (e.g., [5, 236]). This has brought a paradigm shift in computer graphics towards data-driven appearance modeling techniques and databases (e.g., [69, 99, 220]). Although this has allowed us to reach an unprecedented level of realism in visual appearance, *editing* the captured data remains a challenge: First, there is a disconnect between the mathematical representation of the data and any meaningful parameters that humans understand; the captured data is machine-friendly, but not human-friendly. Second, the many different formats and representations require handling potentially hundreds of parameters [10, 41]. And third, real-world appearance functions are usually non-linear and high-dimensional, so editing parameters are rarely intuitive. As a result, visual appearance datasets are increasingly unfit to editing operations, which limits the creative process for scientists, engineers, artists and practitioners in general. In short, there is a gap between the complexity, realism and richness of the captured data, and the flexibility to edit such data.

In this work, we present a novel intuitive control space suitable for a wealth of applications, such as perceptually-based appearance editing for novice users and non-specialists, developing novel appearance similarity metrics, mapping perceptual attributes to analytic BRDFs, or providing guidance for gamut mapping. Given the existence of large databases of measured BRDFs, a seemingly attractive option would be fitting them to parametric models. Unfortunately, this approach does not suit our goal of flexible material editing well, since the error introduced depends on the nature of the BRDF being represented [232]. Moreover, the error metrics that guide such a fitting do not take into account perceptual aspects, which might lead to visible artifacts for seemingly optimal approximations [104]. Last, fitting requires a non-linear optimization which is often numerically unstable, expensive to compute, and typically involves visual inspection to judge the final outcome [232].

Instead, we turn to a non-parametric approach, which can represent with high fidelity a wide scope of measured BRDFs, and lends itself naturally to accommodating our perceptually-based material editing framework. McCool et al. [222] introduced a log-relative mapping that enables a convenient decomposition of measured BRDFs; later Nielsen and colleagues [236] performed a linear decomposition into principal components after this mapping. The first five of these components are nicely descriptive of appearance, but cannot be controlled in an intuitive manner. The reason is twofold: First, as the authors discuss, their components are not able to properly isolate the different effects that characterize appearance; and second, as we will show, linear variations in magnitude of these components result in highly non-linear changes in appearance.

We show that there is a much more intricate correlation between principal components, material appearance, and appearance perception. In our work, we first quadruple the original MERL dataset to 400 BRDFs, by synthesizing novel, mathematically valid samples from measured ones (Sec. 4.3). We then find a mapping between the space of principal components and higher level perceptual attributes that enable intuitive material editing. This is done as follows: First, we perform a series of experiments to obtain a meaningful list of editing attributes (Sec. 4.4.1, Exp. 1). From them, a perceptual rating is obtained from a vast user study in which we gather 56,000 answers, covering all our attributes and BRDFs (Sec. 4.4.2, Exp. 2). We then learn functionals for each of the attributes, mapping the perceptual ratings of each attribute to the underlying principal component basis coefficients (Sec. 4.5.1). These functionals can be readily used to intuitively and interactively edit measured BRDFs, yielding new, plausible appearances (Sec. 4.5.2), as depicted in Figure 4.1.

We validate the *correctness* of our framework through a user study (Sec. 4.8) which shows that our functionals can predict well the attribute values given by users. Further, we also show that it is *intuitive* and *predictable*, as well as *versatile*, allowing for a variety of appearance edits; all this can be found in Sec. 4.8 and Appendix B. Further, and in addition to editing of measured BRDFs, our derived functionals can be used to increase our knowledge on the perception of appearance (Sec. 2.4), and for a number of other applications, described in Sec. 4.7. Finally, to foster further research in this direction, we make both our code and dataset public in the project website[8].

4.2 Related work

EDITING OF PARAMETRIC MODELS These works focus mostly on the interface provided to the user. A paradigmatic example of this is BRDF-Shop [67], where the authors design an artist-friendly editing framework based on an extension of the Ward model. Ngan et al. [233] propose an image-driven navigation over the space with embedded analytical BRDF models, in which the distance between the models is measured as the difference between rendered images of a sphere under natural illumination. Talton et al. [314] develop a collaborative editing system that explores the parameter space of the anisotropic Ashikhmin model [15], based on models saved by other users. Other works focus on fast feedback upon BRDF edits, and treat appearance and lighting jointly [60, 235, 312, 364]. Last, specialized models for car paint design enable BRDF editing by directly specifying the composition of physical paint ingredients, such as density of pigments, or type and distribution of flakes, which affect the appearance of glitter effects [96]. While many of

[8] http://webdiis.unizar.es/~aserrano/projects/Material-Appearance

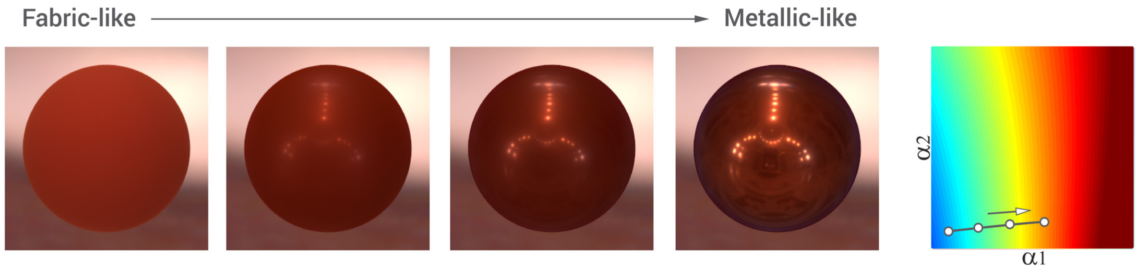


Figure 4.1: Using our control space to achieve fast, intuitive edits of material appearance. We increasingly modify the metallic appearance of a fabric-like BRDF from the MERL database (*red-fabric2*), yielding intuitive changes in appearance by simply adjusting *one* of our perceptual attributes. Key to this ease of use and predictability of the results is our novel functionals, which map the coefficients of the first five principal components (PC) of the BRDF representation to the expected behavior of the perceptual attributes, based on a large-scale user study comprising 56,000 ratings. The rightmost plot shows the path followed by this edit in our control space. Other applications of our novel space include appearance similarity metrics, mapping perceptual attributes to analytic BRDFs, or guidance for gamut mapping.

these techniques support measured BRDFs, the common key obstacle is the lack of a sufficiently general and expressive editing space, which we address in this work.

EDITING OF NON-PARAMETRIC MODELS Editing measured BRDF data without fitting to parametric models is a more challenging task, since the editing space is large and unintuitive [343]. Lawrence et al. [180] proposed the Inverse Shade Trees factorization, which decomposes spatially-varying BRDFs into texture and basis BRDFs, which they further decompose into simple 1D curves representing physical effects. Building on their work, Ben-Artzi et al. [23] proposed a similar framework with precomputed polynomial basis, allowing for complex direct lighting with shadows, as well as interreflections [24]. All these methods lack intuitive parameters, so that editing implies heuristically modifying a set of 1D curves.

INDUSTRIAL STANDARDS A pragmatic approach for a perceptually meaningful characterization of reflectance has been developed by the material industry [146] and formalized in a number of standardization documents by the American Society for Testing and Materials (ASTM). For example, a number of gloss dimensions have been specified [343, Tbl. 1] along with the associated pairs of incident and reflection angles for the reflectance measurements, which should fully characterize the gloss appearance. Westlund and Meyer [337] derive the correspondence between such isolated reflectance measurements and parameters of selected analytic BRDF models, which effectively links them with the industrial characterization of reflectance in terms of gloss, haze, sheen, and other attributes.

PERCEPTUAL EDITING SPACES Many different works have applied perceptual strategies in computer graphics [6, 224]. High dimensional perceptual spaces have been used for style similarity [109], translucency perception [111], interior design taxonomy [22], or shader design [175], to name a few examples. Boyadzhiev et al. [35] introduce a set of intuitive attributes for image-based material editing. Conceptually, the closest methodology to ours has been proposed for garment simulation, although using the parameters of a custom high-quality production pipeline simulator [296]. For BRDF editing, Pellacini et al. [252] observed that a direct parameter tuning for analytic BRDFs is often unintuitive due to strongly non-linear changes in material appearance. By analogy to perceptually uniform color spaces such as CIELAB and CIELUV, they derive a perceptually uniform parameter scaling for the Ward model, which has since been used to study image-driven navigation spaces [233], or the influence of shape in material perception [326]. Wills et al.

[343] extend the concept of perceptually uniform spaces for measured BRDFs, and propose a low-dimensional space suitable for intuitive navigation and construction of new materials, although limited to the achromatic component of reflectance (gloss). Kerr and Pellacini [164] showed that, for the particular task of matching material appearance, the performance of novice users is comparable for the original Ward model and its perceptually linearized version, while image-driven navigation seems to be less efficient. However, the study is limited to colorless BRDFs, and only for two simple sliders: diffuse and specular.

We draw inspiration from the work of Matusik et al. [220], who present a data-driven reflectance model. The authors propose to reduce the dimensionality of measured BRDF data either with linear dimensionality reduction (PCA) or with non-linear dimensionality reducers, resulting in a 45D or 15D (respectively) manifold. Then, they define a set of *perceptual traits* (such as redness or silverness), and have a *single* user perform a binary classification whether a given material possesses each particular trait or not. Trait vectors enable navigation in their BRDF spaces by specifying the directions of desirable changes for a given trait or their combinations. Our work is different in many ways: we emphasize on a perceptually meaningful material characterization, but employ a set of carefully selected attributes, which have been identified in a large-scale experiment as intuitive, descriptive, and discriminative when describing reflectance properties. We inherit a perceptually meaningful scaling and decomposition of raw BRDF data akin to perceived contrast, which greatly reduces PCA dimensionality [236], making it comparable to purely perceptually derived spaces [343]. We perform dense uniform sampling of the scaled PCA space, synthesizing additional BRDFs from the initial MERL dataset (totaling 400), and obtain ratings for our perceptual attributes in another large scale experiment from which we collect over 56,000 answers from 400 participants. This allows us to reconstruct perceptually-based complex embeddings of our attributes in the PCA-space, which enables intuitive, predictable, and interactive appearance changes from measured BRDF data.

4.3 BRDF representation and database

4.3.1 *Principal components space*

A database of measured BRDFs can be used to learn a principal components (PC) basis, in which any other BRDF can be represented as [220, 233, 236]:

$$\mathbf{b} = \mathbf{Q}\boldsymbol{\alpha} + \boldsymbol{\mu} \tag{4.1}$$

where $\mathbf{b} \in \mathbb{R}^N$ is the BRDF represented in the basis, $\mathbf{Q} \in \mathbb{R}^{N \times M}$ is the matrix representing the PC basis (specifically, the eigenvectors of the basis scaled by their eigenvalues), $^- \in \mathbb{R}^N$ is the average of the measured data, and $\boldsymbol{\alpha} \in \mathbb{R}^M$ are the coefficients of each of the components for the particular BRDF $\mathbf{b}$.

Similar to other approaches [222, 236], we perform a log-relative linear mapping of the reflectance data, to avoid allocating most of the available dynamic range to encode variations in the specular peak, and represent our BRDFs with the resulting first five principal components (i.e., $M = 5$), which are loosely related to some characteristics of material appearance [236]. To make sure that working in a reduced space does not affect the perception of appearance, we have run a small-scale experiment with 20 BRDFs from the MERL database, covering a wide range of appearances. We followed a 2AFC approach, showing the original BRDF, and our representation with only the first five components, and asked the (49) participants to choose which of the two shown images better conveyed a given attribute. The χ^2 analysis of the results (see Appendix B) showed that participants were mostly selecting at random, which indicates that our five dimensional space does not degrade appearance perception and is suitable for our purposes. Examples of the stimuli are shown in Figure 4.2, while details on methodology, analysis, and results appear in Appendix B. Moreover, limiting the space to five dimensions has an additional advantage:

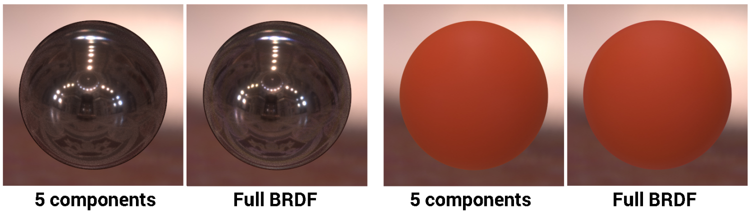


Figure 4.2: Examples of the stimuli used in our pilot test to determine whether working in a reduced space affects the perception of appearance (shown are *ss440* and *dark-red-paint*, both from the MERL database). The analysis of the results indicates that a five-dimensional space is sufficiently descriptive for our purposes.

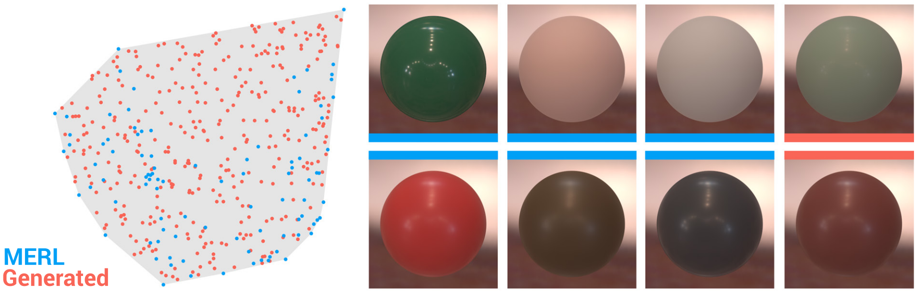


Figure 4.3: *Left:* A two-dimensional projection of our 5D BRDF convex hull. Blue dots represent projections of the original MERL BRDFs, while the red dots represent our newly generated samples. Our Gibbs sampling strategy ensures a good coverage of the PC space. *Right:* Each row shows three original BRDFs (blue), plus a novel synthesized material derived from them (red).

When creating a larger database of BRDFs (Sec. 4.3.2), it helps improve the sampling process by avoiding regions with little impact on appearance.

However, this 5D space, while sufficiently descriptive for our purposes, does not lend itself to intuitive material editing, since the dimensions do not clearly correlate with isolated material properties: Some components (such as the fifth component) are responsible for a combination of different effects, while other effects (such as the shape of the specular highlights) depend on several components. *This suggests that there is a much more intricate correlation between principal components, material appearance, and appearance perception*, as we will show. Moreover, our goal is to be able to modify appearance based on higher-level attributes (such as *glossy*, or *plastic-like*), which do not have a direct, one-to-one correlation with PCA components.

4.3.2 *Creating a database of BRDFs*

In order to get a sufficiently large amount of data to gather perceptual ratings of our attributes and build the mappings, we need to work with a large BRDF dataset, offering a varied and adequately sampled range of materials with different perceptual properties. Acquiring such a large dataset would be both time consuming and challenging; instead, we opt to synthesize novel BRDFs from existing ones. We choose the MERL database [220] as a starting point, which consists of 100 homogeneous materials that cover reasonably well the range of real world isotropic materials. Similar to their original work, we remove materials with visible non-homogeneities, anisotropy or subsurface scattering, ending up with an initial seed of 94 materials, from which we will synthesize new ones.

It has been shown that the space of mathematically valid BRDFs is convex [219, 343]. This means that any convex combination of two given BRDFs will produce a new one where non-negativity, energy conservation and reciprocity are preserved. While not all possible combinations will produce a material likely to be found in the real-world, in our work we favor intuitive, *artistic* exploration and expression of material appearance. We therefore compute the convex hull of the measured BRDFs of the MERL database projected in our five-dimensional PCA space, with the goal of synthesizing novel BRDFs inside the polytope defined by the 5D convex hull. When synthesizing our new BRDFs, we aim to achieve a close-to-uniform coverage along our PC dimensions, so that the full space is well represented in our perceptual experiments. Since the exact calculation of a high dimensional polytope is computationally expensive, we choose to approximate a uniform distribution with Gibbs sampling [225] within the convex hull (see Figure 4.3, left).

Then, for each sample, we synthesize a novel BRDF as a convex combination of the three nearest original MERL BRDFs, weighted by their distances to the sampled point. Note that the sampling and the distances are computed in a five-dimensional space, but the convex combination that leads to the novel BRDF is performed on a per-channel basis in a 15-dimensional space (5 x 3 color channels). Figure. 4.3 (right) shows two novel BRDFs synthesized this way. We generate with this method 306 new BRDFs, yielding a total of 400 different materials for our tests, which can be found in the project website.

4.4 Experiments

We ran a first test to build a user-friendly, intuitive set of attributes for appearance editing; for the sake of conciseness, we only briefly summarize here the main results. In a second test, we obtain a perceptual rating of those attributes, which will allow us to build a mapping between the attributes and the underlying PCA basis coefficients. Please refer to Appendix B for additional details, including a full description of our first experiment.

4.4.1 *Experiment 1: Building the space of attributes*

For this first test, we rendered a large number of stimuli depicting different materials, built an extensive initial list of candidate appearance descriptors, and then relied on a user study to reduce them to a suitable size. Inspired by recent works on material perception and design (e.g., [152, 164]), our stimuli consist of spheres of 60 different materials from the MERL database [220], chosen to span a wide range of different appearances, and lit by direct illumination. Our initial list was made up of 28 attributes, ranging from high level class descriptors (e.g. ceramic-like) to low level appearance descriptors (e.g., strength of reflections). Relying on Fleming's work [101], where he states that *we can also make many judgments about the perceived qualities of different materials irrespective of their class membership*, we do not make any restrictions about the type of descriptors in our list. The final list consists of fourteen attributes, covering both high- and mid-level features: *plastic-like, rubber-like, metallic-like, fabric-like, ceramic-like, soft, hard, matte, glossy, bright, rough, tint of reflections, strength of reflections*, and *sharpness of reflections*.

4.4.2 *Experiment 2: Measuring the attributes*

Once we have built a suitable list of perceptual attributes, our next goal is to characterize a large number of materials based on such a list, which will allow us to derive mappings between attributes and the underlying basis coefficients of the BRDFs. We obtained a total of 56,000 rating responses (400 BRDFs × 10 responses/BRDF × 14 questions/BRDF), which we will use to build the mappings between the perceptual attributes and the underlying PCA coefficients, as described in the next section.

STIMULI To increase the variability of the analyzed BRDFs, we significantly extended our stimuli from the previous experiment, including all our 400 different materials, generated as described in Sec. 3.2. The materials are rendered with PBRT, using the *St. Peter's* environment map. This is also the case for Exp. 1: details on this choice can be found in Appendix B

PARTICIPANTS Since we aimed to gather a very large number of answers, we followed similar large-scale studies in computer graphics (e.g., [34, 270]) and used Amazon Mechanical Turk[9]. A total of 400 paid subjects took part in our experiment, casting a total of 56,000 rating votes. The feedback we received through the online platform was very positive: they enjoyed the test, and found it engaging and interesting.

PROCEDURE To analyze how different materials are characterized in terms of our list of perceptual attributes, we first considered different options. A valid alternative in principle would be a double-stimulus method, such as a forced-choice pairwise comparison. In such scenario, a ranking (ordering) task could be devised [54, 245], which is easy for the participants, and usually results in low variance in their responses; however, as Yumer et al. show [348] a rating approach may be better suited for multi-modal problems like ours, where different BRDFs may have similar attribute strengths. On the other hand, methods to derive a meaningful perceptual scaling from pairwise ranking data exist [297]. Unfortunately, they require close stimuli placement with small attribute differences (ideally overlapping in terms of JNDs), in order to avoid consistent responses where all the votes go to the same stimulus. The lack of a distribution of the user responses might indicate a suprathreshold difference, and does not provide any useful information on attribute scaling. Such a careful placement of the stimuli typically requires extensive pilot studies that would not be practical given the large number of attributes and the 5D embedding that we consider in this work. Another option would be rating pairwise stimuli [175, 348]. While this leads to better scaling properties than ranking, it would substantially increase the number of trials, making the tests impractical. Typically a random subset of pairs is considered; only when the parameter space is known, nearby pairs can be selected. (e.g., most of the images lack the rubber-like attribute in our case).

While different pros and cons for each approach can be observed, it has been recently reported after extensive tests that there is no evidence that double stimulus methods are more accurate than single stimulus methods [211, 320]. Taking this into account, and in light of the analysis above, we therefore rely on magnitude estimation through rating, also referred to as Mean Opinion Score (MOS). This single-stimulus approach is a well-established methodology, dominant in image and video experiments, and recommended by standard international organizations such as ITU or ISO [147, 148, 162].

Similar to previous works [89, 355], we chose a five-point scale, which we found offered a good trade-off between the number of options and the difficulty to carry out the test. Each scale was numbered from 1 (*none, or very little*) to 5 (*a lot*). We designed a web-based interface, for easy navigation. The participants' task and the rating scales were explained at the beginning, before proceeding to the actual test. During the test, the participants were shown one rendered material at a time, plus the fourteen perceptual attributes from Exp. 1; they were asked to rate each of the perceptual attributes, for each BRDF, in the Likert-type scale (a screenshot of the test can be seen in Appendix B). We thus obtained the 56,000 rating responses, which we will use to build the mappings between the perceptual attributes and the underlying PCA coefficients, as described in the next section.

4.5 An intuitive appearance control space

We now describe how we build our mapping between each attribute and the coefficients of the five PCs defining a BRDF, based on the ratings obtained. These mappings will define our intuitive control space for appearance editing.

[9] Herr and Bostok [136] recently demonstrated the viability of crowdsourcing graphical perception studies, reducing variance and finding a good match with results from classic experiments.

4.5.1 *Fitting functionals for the attributes*

Similar to related works [252, 343], we decouple achromatic reflectance from color information (working in CIELab space), which adds flexibility to our editing framework. Our functionals are derived for achromatic reflectance, but changing chromaticity can be easily accomplished by modifying the a and b channels, as shown in Figure 4.4. For each of the attributes, we seek a functional $\varphi : \mathbb{R}^5 \rightarrow \mathbb{R}$ that models the behavior of the attribute as a function of the coefficients of the first five PCA components $\boldsymbol{\alpha} = \{\mathbf{s}_1..\mathbf{s}_5\}$. We will obtain these functions using as input data the ratings given by subjects to each of the BRDFs in Exp. 2, and the $\boldsymbol{\alpha}$ coefficients of those BRDFs. A priori we have no information of how these functions should look like. We therefore choose to use a radial basis function (RBF) network with one hidden layer; RBFs are known to be capable of providing approximations to any continuous function of a finite number of real variables with arbitrary precision [247], and have been used before to approximate BRDFs [361]. Our RBF network can be expressed as follows:

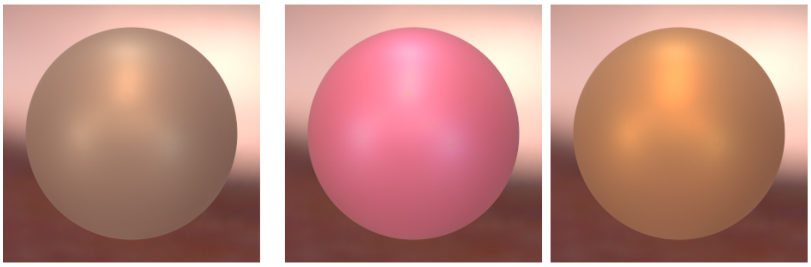


Figure 4.4: We can modify chromaticity in our framework simply by tuning the chromaticity channels of the CIELab space. *Left:* original BRDF. *Right:* two new BRDFs, generated as described in the text.

$$y = \varphi(\boldsymbol{\alpha}) = \sum_{i=1}^{N_c} \theta_i \phi(\|\boldsymbol{\alpha} - \mathbf{c}_i\|) \tag{4.2}$$

where $y \in \mathbb{R}$ is a value, in the range [0..1], that represents the strength of the attribute, N_c is the number of neurons in the network, $\mathbf{c}_i \in \mathbb{R}^5$ are the centers of such neurons, and θ_i the weights of each neuron. Following common practice, we model ϕ as a Gaussian function, and define the norm as the Euclidean distance:

$$y = \varphi(\boldsymbol{\alpha}) = \sum_{i=1}^{N_c} \theta_i \exp^{-\beta\|\boldsymbol{\alpha} - \mathbf{c}_i\|^2} \tag{4.3}$$

where β controls the smoothness of the Gaussian functions.

For each attribute, we train a network by minimizing the L2-norm between the average attribute values given by subjects for each BRDF (MOS), $y_j \in \mathbb{R}^{N_b}$, where $N_b = 325$ is the number of BRDFs used for training, and the value of $\varphi(\mathbf{x}_j)$. The neuron centers are found by k-means clustering. When choosing the number of neurons N_c, we must find a compromise: Too few neurons will not provide enough degrees of freedom to capture the complexity of the space, while having too many has the risk of overfitting the data. We measure the fitting error (MSE) for several values of N_c, as well as for neuron-varying values of β, as shown in Figure 4.5 (left), and choose $N_c = 10$ neurons with uniform β, since larger values of N_c or non-uniform β values do not offer a significant decrease in error.

Additionally, we evaluate the goodness-of-fit of the RBFs by calculating, for each attribute and for all the BRDFs in our database, the mean distance between the values predicted by our functionals, and the answers given by each particular user; we plot them by projecting them to a 2D slice ($\alpha_1 - \alpha_2$) of our space. In Figure 4.5 (right) we plot these distances for the *rubber-like* attribute (plots for all the attributes are available in the project website). The low values indicate that our

RBFs fit users' opinions well in all regions of the space. Two important conclusions can be derived: First, our RBFs fit the data well in all the regions of the space; second, the high agreement confirms that we can use the MOS as a good approximation of users' opinion. Note that these distances are indirectly indicative of confidence (in the sense of agreement between users): If the variance in users' responses for an attribute and BRDF were high, then the mean distance values plotted here would also be high. Nevertheless, we specifically look into user agreement in Sec. 4.6.2.

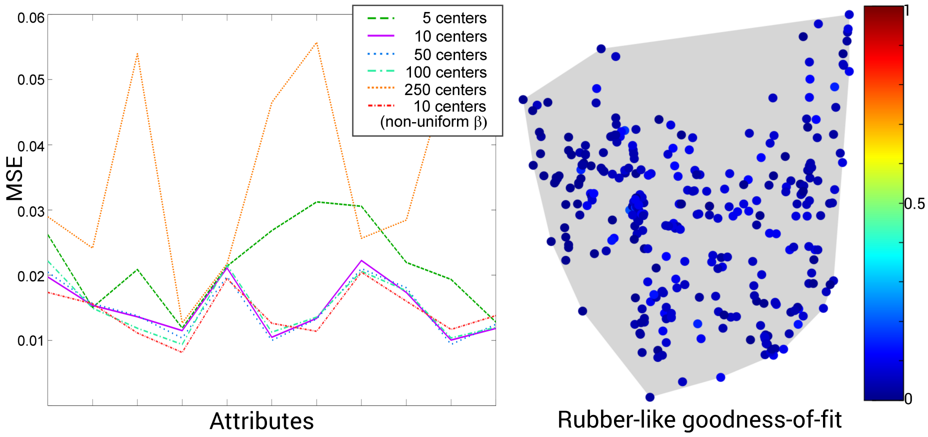


Figure 4.5: *Left:* Fitting error (MSE) for our attributes (x-axis). The curves correspond to different numbers of neurons of the RBFs and for non-uniform β values (neuron-varying smoothness of the Gaussians); we choose 10 neurons. *Right:* Mean distances between values predicted by our functionals and subjects' answers for the attribute *rubber-like* for each BRDF. The low values indicate that our RBFs fit users' opinions well in all regions of the space, and confirm the adequacy of the use of the MOS in their computation.

4.5.2 *Navigating the space with perceptual attributes*

Once we have the functionals mapping a set of coefficients $\mathbf{s}_i$ to attribute values y, we can interactively navigate our control space. To train our functionals, we map the first five PCA components to each of the attributes individually: $\mathbb{R}^5 \to \mathbb{R}$, obtaining one mapping per attribute. This mapping is surjective (though not bijective). However, for certain applications (e.g., editing) we need the inverse: given an attribute value y, find the corresponding coefficients $\mathbf{s}_i$. The solution to this inverse problem is not unique; we formulate it as a minimization, which we solve via gradient descent (this will be demonstrated in Sec. 4.7).

Starting with any BRDF, we compute its $\boldsymbol{\alpha}$ coefficients by projecting it into the PCA basis, and subsequently the values y_k associated to each of the attributes of interest using the mappings $\varphi_k : \mathbb{R}^5 \to \mathbb{R}$, as explained in Sec. 4.5.1. Given an attribute k, and an initial BRDF with coefficients $\boldsymbol{\alpha}_{ini}$, we can alter appearance by modifying its initial value $y_{k,ini}$ to an objective value $y_{k,obj}$. We formulate this as a minimization (we drop k for clarity):

$$\min_{\boldsymbol{\alpha}} \left\| \varphi(\boldsymbol{\alpha}) - y_{obj} \right\|^2 \tag{4.4}$$

where $\varphi(\cdot)$ is given by Equation 4.3. We solve this using gradient descent with $\mathbf{x}_0 = \mathbf{x}_{ini}$, aiming to obtain the solution closest to $\boldsymbol{\alpha}_{ini}$ (i.e., the initial BRDF) that satisfies the requirement for y_{obj}. To ensure that the solution remains within the five-dimensional convex hull defined by the measured MERL BRDFs, we test at each step whether the new location remains within its boundaries, and stop the minimization if the boundary is reached. Note that, given the correlation between perceptual attributes, changes in one may affect other attributes as well, to reflect a perceptually valid change in material properties. Since our functionals are derived from perceptual ratings given by users, their values will be intuitive and correlate with user perception, as we validate in Sec. 4.8.

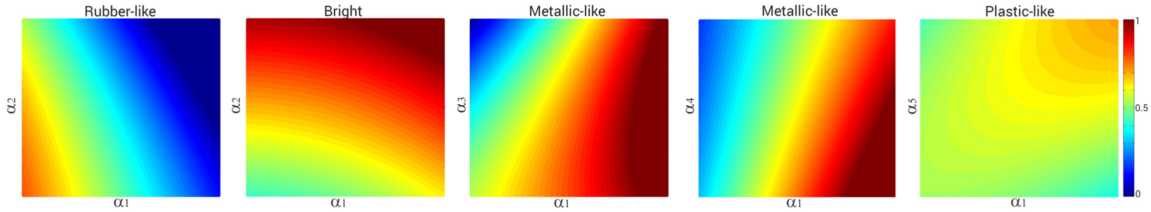


Figure 4.6: Sample 2D slices of our functionals $\varphi : \mathbb{R}^5 \rightarrow \mathbb{R}$, mapping coefficients $\boldsymbol{\alpha}$ in the PC basis to perceptual ratings for different attributes and along different dimensions. *From left to right: rubber-like* ($\mathbf{s}_1$-$\mathbf{s}_2$ slice); *bright* ($\mathbf{s}_1$-$\mathbf{s}_2$ slice); *metallic-like* ($\mathbf{s}_1$-$\mathbf{s}_3$ slice); *metallic-like* ($\mathbf{s}_1$-$\mathbf{s}_4$ slice); and *plastic-like* ($\mathbf{s}_1$-$\mathbf{s}_5$ slice). Please refer to text for further interpretation.

4.6 Analysis and exploration of the space

In this section we first provide a qualitative analysis of our functionals. We then analyze the different attributes, the interactions between them, and the agreement between user responses for different attributes and BRDFs. Finally, we explore the correlation between our attributes.

4.6.1 *Qualitative analysis of the attribute functionals*

In our work, we map the space of principal components to higher level perceptual attributes that define an intuitive control space for appearance; these mappings will then be used to find the *paths* in PC space that lead to natural-looking appearance changes. Figure 4.6 shows a series of 2D slices of our 5D space, defined by the coefficients $\mathbf{s}_i$, for different material attributes depicting our mappings using our functionals. We plot two-dimensional slices $\mathbf{s}_1 - \mathbf{s}_i$ ($i = 2..5$), since the first component $\mathbf{s}_1$ has the greatest influence on material appearance. A qualitative analysis reveals interesting insights that align well with our intuition of how we perceive some characteristics of materials. As we explain below, observations on two-dimensional slices of our 5D PC space confirm that: i) analyzing each principal component of the BRDFs in isolation cannot explain how materials are perceived; instead, there are many correlations defined in our larger five-dimensional space; and ii) our approach correlates well with human perception of materials, since we find many expected behaviors in our two-dimensional projections. In the following we describe the different slices in Figure 4.6:

- The first slice depicts how the *rubber-like* attribute varies with *both* $\mathbf{s}_1$ and $\mathbf{s}_2$ (the specular and diffuse components, respectively). High values of both the specular and diffuse coefficients yield low values for perceived rubber-like, and viceversa. Moreover, as the specular intensity $\mathbf{s}_1$ increases, the material becomes less rubber-like, while as the diffuse component increases $\mathbf{s}_2$, the material also loses its rubber-like look. This is consistent with our intuition that rubber-looking materials do not show specular highlights and reflect relatively little light overall.

- The second slice analyzes again the $\mathbf{s}_1$-$\mathbf{s}_2$ plane for *bright*, and shows how both coefficients have an influence on how bright a material looks. Although mainly dominated by $\mathbf{s}_2$ (increased brightness is correlated with an increase in the diffuse component), $\mathbf{s}_1$ also plays a role: For a fixed value of $\mathbf{s}_2$, increasing the specular component also causes the perceived brightness to increase.

- The third slice corresponds to the *metallic-like* attribute, and in this case depicts an $\mathbf{s}_1$-$\mathbf{s}_3$ (both related to the specular component) cut of the 5D space. For low and mid values of the intensity component $\mathbf{s}_1$, the component related to the shape of the specularities $\mathbf{s}_3$ plays a significant role: materials appear more metallic as its value decreases. However, for very high values of the intensity, the shape of the specular highlights becomes increasingly irrelevant when identifying the material as metallic.

- In the fourth slice we study again the *metallic-like* attribute, this time as a function of $\mathbf{s}_1$ and $\mathbf{s}_4$ (Fresnel). As expected, the specular component $\mathbf{s}_1$ dominates the metallic look; but we can clearly see an interesting effect: given a value of $\mathbf{s}_1$, the perceived metallic quality of the material increases as the Fresnel effect $\mathbf{s}_4$ decreases.

- In the last slice, we plot how the *plastic-like* attribute varies with the coefficients $\mathbf{s}_1$ and $\mathbf{s}_5$. A material is more plastic-like as its specular intensity ($\mathbf{s}_1$) increases, as expected; however, the shape of the specular and the Fresnel effect, partially controlled by $\mathbf{s}_5$, also play an important role.

4.6.2 *Inter-user and intra-cluster agreement*

We cluster the measured BRDFs manually into one of six groups according to the *actual* material they belong to, namely *fabric*, *metallic*, *acrylic*, *plastic*, *phenolic*, and *metallic-paint*. We use only measured BRDFs since they can be clustered reliably, following Matusik's naming system [220]. We now seek to analyze the agreement between users when rating each attribute, as well as the agreement between BRDFs from the same material cluster (i.e., whether they share the same appearance).

We obtain, for each cluster and attribute, the *mean score* and a measure of *agreement*. Figure 4.7 shows the resulting plots for a sample cluster; the complete plots for all the clusters can be found in Appendix B. These plots give us a large amount of information about subjective BRDF appearance; in the following, we describe the interpretation of these plots, and present some of the main conclusions.

MEAN SCORE PLOTS For the mean score we compute the mean value per BRDF per attribute, and box plots showing the interquartile range (IQR, defined as Q3-Q1), and maximum and minimum values (Figure 4.7, left). The mean values indicate the general trend of the attribute in the cluster (note that the y-axis is normalized). As with the correlation analysis, the results align with real-world experience; for instance, *metallic-like*, *glossy*, and the *strength* and *sharpness of reflections* all have high mean values for the *metallic* cluster, and much lower for *fabric*. Low variance of the mean for one attribute indicates that such an attribute is a potentially good descriptor of the cluster, while high variance indicates that it is not, since different BRDFs in the cluster are given very different values for such an attribute. For instance, *rough* is not a good descriptor of the *plastic* cluster (see Appendix B), which makes sense since plastic materials can have a wide range of surface roughnesses. As a consequence, a consistently high variance of the mean for multiple attributes in a cluster indicates that users do not identify it as a cluster of appearance; this is the case with plastic BRDFs, probably because they can exhibit a wide variety of appearances in the real world. Finally, note that a low IQR (small box plot) in the mean scores indicates that most of the BRDFs in the cluster share the same average value of the attribute, but not necessarily that users agreed when grading such an attribute for each BRDF; instead, it is the agreement box plots that give an indication of user agreement.

AGREEMENT PLOTS As a measure of agreement we compute the variance of the scores per BRDF per attribute, and box plots showing the mean of this variance, together with its IQR and maximum and minimum values (Figure 4.7, right, and Appendix B). Overall, our plots show consistently low mean values, indicating a large agreement for all clusters and attributes (note that although the maximum value the variance can take is one, the y-axes of the agreement plots range only from 0 to 0.25 for visualization purposes). This suggests that our choice of attributes is adequate for our purposes, being meaningful and intuitive descriptors of appearance; moreover, it also validates using the MOS for the fitting of the functionals. Additionally, a low IQR indicates a good agreement between users for all BRDFs in the cluster, independent of whether the value for the given attribute was high or low (see for instance *glossy* in the *plastic* cluster). A high IQR indicates that for some BRDFs there is agreement, but for others there is not (such as *rubber-like* in the *acrylic* cluster, Figure S.5).

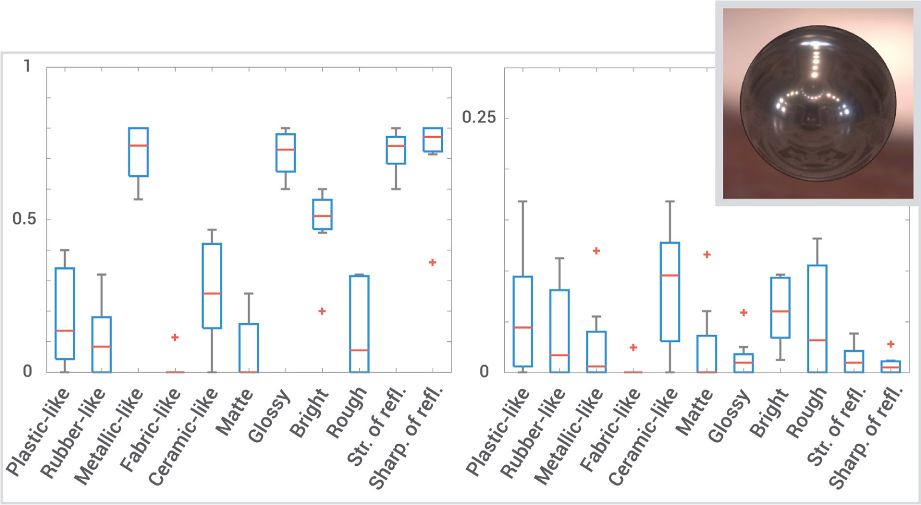
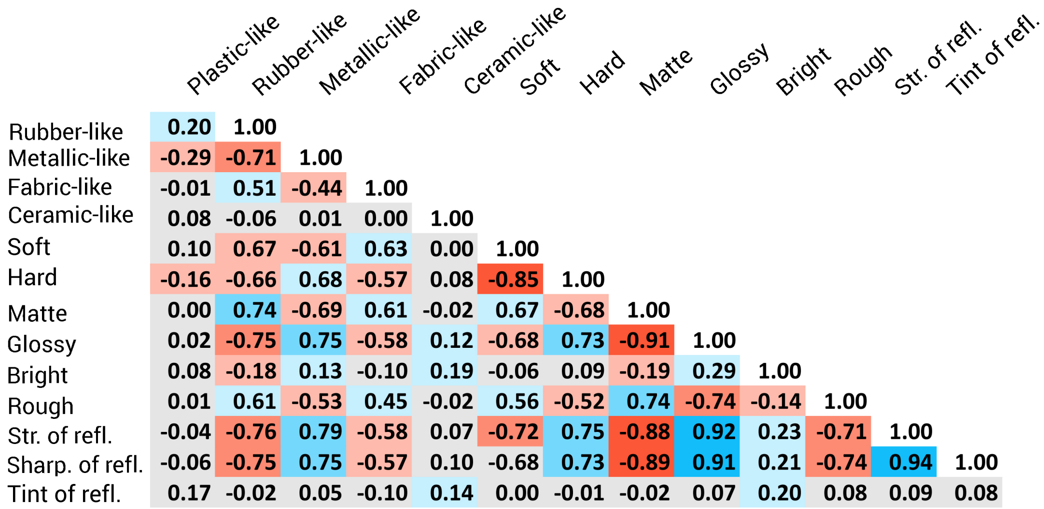


Figure 4.7: Mean scores (left) and agreement (right) for BRDFs in the *metallic* cluster. The inset shows a sample BRDF from the cluster (*aluminium*). A low mean value in the plot on the right is indicative of a high agreement between users; please see the main text for further explanations.



	Plastic-like	Rubber-like	Metallic-like	Fabric-like	Ceramic-like	Soft	Hard	Matte	Glossy	Bright	Rough	Str. of refl.	Tint of refl.
Rubber-like	0.20	1.00											
Metallic-like	-0.29	-0.71	1.00										
Fabric-like	-0.01	0.51	-0.44	1.00									
Ceramic-like	0.08	-0.06	0.01	0.00	1.00								
Soft	0.10	0.67	-0.61	0.63	0.00	1.00							
Hard	-0.16	-0.66	0.68	-0.57	0.08	-0.85	1.00						
Matte	0.00	0.74	-0.69	0.61	-0.02	0.67	-0.68	1.00					
Glossy	0.02	-0.75	0.75	-0.58	0.12	-0.68	0.73	-0.91	1.00				
Bright	0.08	-0.18	0.13	-0.10	0.19	-0.06	0.09	-0.19	0.29	1.00			
Rough	0.01	0.61	-0.53	0.45	-0.02	0.56	-0.52	0.74	-0.74	-0.14	1.00		
Str. of refl.	-0.04	-0.76	0.79	-0.58	0.07	-0.72	0.75	-0.88	0.92	0.23	-0.71	1.00	
Sharp. of refl.	-0.06	-0.75	0.75	-0.57	0.10	-0.68	0.73	-0.89	0.91	0.21	-0.74	0.94	1.00
Tint of refl.	0.17	-0.02	0.05	-0.10	0.14	0.00	-0.01	-0.02	0.07	0.20	0.08	0.09	0.08

Figure 4.8: Color-coded correlation matrix between attributes (Pearson correlation coefficients): Grey, blue and orange indicate no significant correlation ($p - value > 0.05$), positive, and negative correlations, respectively. Darker shades indicate increasingly stronger correlation, see text for details.

4.6.3 *Correlation between attributes*

We finally analyze the *semantic similarity* of our attributes. Figure 4.8 shows Pearson correlation coefficients for our attribute set[10]. Grey color indicates no significant correlation ($p - value >$

[10] We obtained similar results in terms of trends and significance using Spearman rank correlation; they can be found in Appendix B.

0.05), while blue and orange indicate positive and negative correlations, respectively. We have additionally highlighted with increasingly stronger shade those pairs showing a larger correlation (three levels, delimited by > 0.7 and > 0.8 in absolute value). The results match what we would expect: For instance, the *strength* and the *sharpness* of reflections are highly correlated with the perception of how *glossy* a material is, but inversely correlated to how *matte* a surface looks. Similarly, *rough* is highly correlated with *matte*, whereas *rough* and *sharpness* of reflections, or *hard* and *soft*, show a strong negative correlation.

This shows the overall correlation between our functionals, but we can further analyze locally in which particular regions of the space our attributes have a similar behavior: Since our underlying five-dimensional space is of lower dimensionality than the number of attributes, there may be regions in which multiple attributes exhibit the same behavior. In the top part of Figure 4.9, we analyze two attributes: *strength of reflections* and *metallic-like*. In each row, we show $\mathbf{s}_1 - \mathbf{s}_2$ plots for low and high values of $\mathbf{s}_3$, $\mathbf{s}_4$, and $\mathbf{s}_5$. These plots show that both attributes highly depend on $\mathbf{s}_1$; this agrees with our intuition, since $\mathbf{s}_1$ roughly corresponds to the specular trait of the material. But the plots also reveal that for varying values of $\mathbf{s}_3..\mathbf{s}_5$, this dependency does not change much for *strength of reflections*, indicating that $\mathbf{s}_1$ dominates the value of the attribute in the whole 5D space. The *metallic-like* attribute, however, has a more complex behavior, showing a larger dependency on $\mathbf{s}_2$ and $\mathbf{s}_3$ in the region of the 5D space defined by low $\mathbf{s}_3$ values (see the top-left plot). Despite this difference in this particular region, the overall dependency on $\mathbf{s}_1$ translates into the high correlation shown in Figure 4.8. In the bottom row of Figure 4.9, we further analyze a second pair: *strength of reflections* and *rubber-like*. The latter shows a large dependency on $\mathbf{s}_1$ in the regions of the space defined by high values of $\mathbf{s}_3..\mathbf{s}_5$; this dependency is almost the opposite with respect to *strength of reflections*, as one would expect (the stronger the specular, the less rubber-like it appears). In addition, *rubber-like* shows a very different behavior in the regions where $\mathbf{s}_3..\mathbf{s}_5$ have low values, indicating that $\mathbf{s}_2$, related to the diffuse component, plays a crucial role in our perception of rubbery appearance in certain cases. In particular, for low values of $\mathbf{s}_3$, indicative of a high roughness, the relative importance of $\mathbf{s}_1$ decreases in favor of $\mathbf{s}_2$; this is also the case for the metallic-like attribute explained before.

Figs. 4.8 and 4.9 also show that our different perceptual attributes are not orthogonal. This was already observed by Matusik et al. [220], while a similar conclusion was reached for perceptual parameters designed for garment simulation [296]. This is a desirable characteristic for a control space, since it prevents the user from trying to produce a BRDF that is *glossy* and *matte* at the same time, for instance.

4.7 Applications

MATERIAL EDITING As we have seen, our mappings reveal complex relationships between PC components and appearance attributes, so intuitive edits cannot be performed directly on the PC components. Moreover, since our mappings are derived from perceptual ratings, changes in the attributes are more predictable and intuitive than changes in the PC coefficients. We illustrate this in Figure 4.10: the top row shows a linear interpolation from a fabric-like BRDF to a metallic-like BRDF in the log-relative mapped PC-space of Nielsen et al. [236]; the bottom row shows such a transition using our perceptual attributes. Our transitions look more equally spaced in terms of appearance, while moving linearly in the PC-space yields sudden and non-linear changes in appearance. Figure 4.11 shows more edited BRDFs using our framework, obtained from measured BRDFs from the MERL database. For each original BRDF, we linearly vary the value of one of our perceptual attributes, and render the resulting BRDF at each step. The resulting BRDFs are obtained using the procedure described in Sec. 4.5.2. As the figure shows, our user-friendly editing space yields feasible and appealing edited BRDFs, while keeping variations perceptually meaningful. The last row shows the path followed through our 5D space while varying each of the attributes (we show the most representative 2D slice). Additionally, in Figure 4.12 we make significant changes in the appearance of the teapots, for input BRDFs very different in nature. All

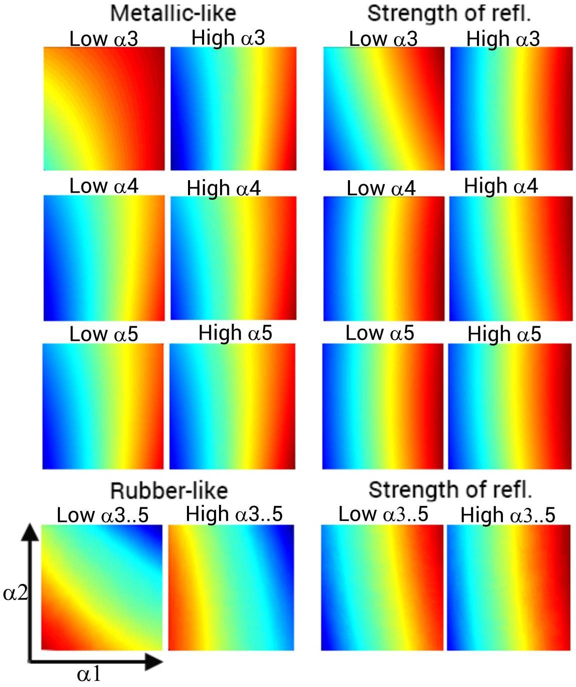


Figure 4.9: Comparison of the behavior of three attributes in different regions of our five-dimensional space. Attributes are: (i) *metallic-like*, (ii) *strength of reflections*, and (iii) *rubber-like*. (i) and (ii) exhibit a high positive correlation, so they behave similarly in most regions of the space. On the contrary, (ii) and (iii) exhibit a high negative correlation, behaving very differently in most regions. Colormaps depict the value of the corresponding attribute in the $\mathbf{s}_1 - \mathbf{s}_2$ plane of our 5D space for different values (low/high) of the remaining coefficients ($\mathbf{s}_3..\mathbf{s}_5$).

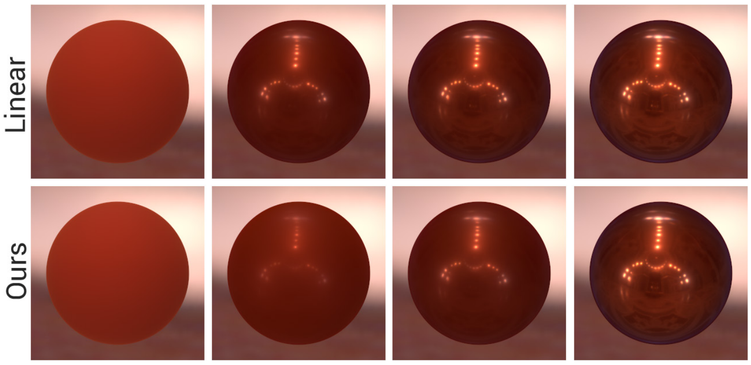


Figure 4.10: Comparison between linear interpolation in the log-relative mapped PC-space of Nielsen et al. [236] and traversing our perceptually-based space, going from a fabric-like to a metallic-like BRDF. Our edits are more perceptually-uniform, whereas a linear interpolation in PC-space causes sudden, unpredictable changes in appearance.

the edits have been achieved by tuning *a single attribute* in our control space. More edited BRDFs with different illuminations can be found in the project website.

SIMILARITY METRICS Similarity metrics are a useful tool to determine if two images are visually equivalent, i.e., if they convey the same impressions of scene appearance [258]. Establishing a measure of similarity between two BRDFs would be very useful for a large number of applications, including gamut mapping, BRDF compression, fitting, or even acquisition. Different metrics have been proposed, such as root mean squared error (RMSE) and its variants (cosine-weighted, with or without cubic root) [104], or the perceptually uniform reparametrizations of analytic BRDF spaces [252]. However, the definition of a global similarity metric does not allow to analyze perceptual attributes *separately*: Two BRDFs could have very similar specular peaks, yet strikingly different diffuse properties. Our functionals offer a novel approach, providing a means for evaluating similarity for *individual visual attributes*. Figure 4.13 shows an example with three BRDFs (A, B, and C). Pairwise comparisons (A-B, and A-C) using an RMSE similarity metric [104] yield very similar results (6358 vs. 6365), although it seems obvious to think of A-B as more similar than A-C. Instead, our functionals allow to break down the notion of similarity in terms of specific aspects of appearance. For instance, in terms of brightness, our metric accurately yields a much closer distance between A-B (0.0452), than A-C (0.5171).

PERCEPTUAL ATTRIBUTES FOR ANALYTIC BRDFS Our methodology, together with the subjective data compiled in our user studies, can be used to derive novel functionals relating our perceptual attributes to other BRDF representations, such as analytic models. We fit our database (the 400 BRDFs) to a chosen model, and then use the answers collected in our user study to train new functionals relating the set of perceptual attributes to the parameters of the analytic model. Figure 4.14 demonstrates this for the *blue-fabric* BRDF from MERL's database, fitted to a microfacet model with a Beckmann distribution. The middle image shows the learned functional for the chosen attribute. We can see how the attribute's value varies non-linearly with the parameters k_s (specular) and A (roughness) of the model: For low roughness the influence of k_s in the strength of reflections is much larger, as one would expect. Our functional hides the complexity of the parameters' interactions in the original representation, making the editing process more predictable and intuitive. Since our extended database will be made public, this will facilitate the learning of new functionals for any other BRDF representation, as well as the exploration of the resulting spaces.

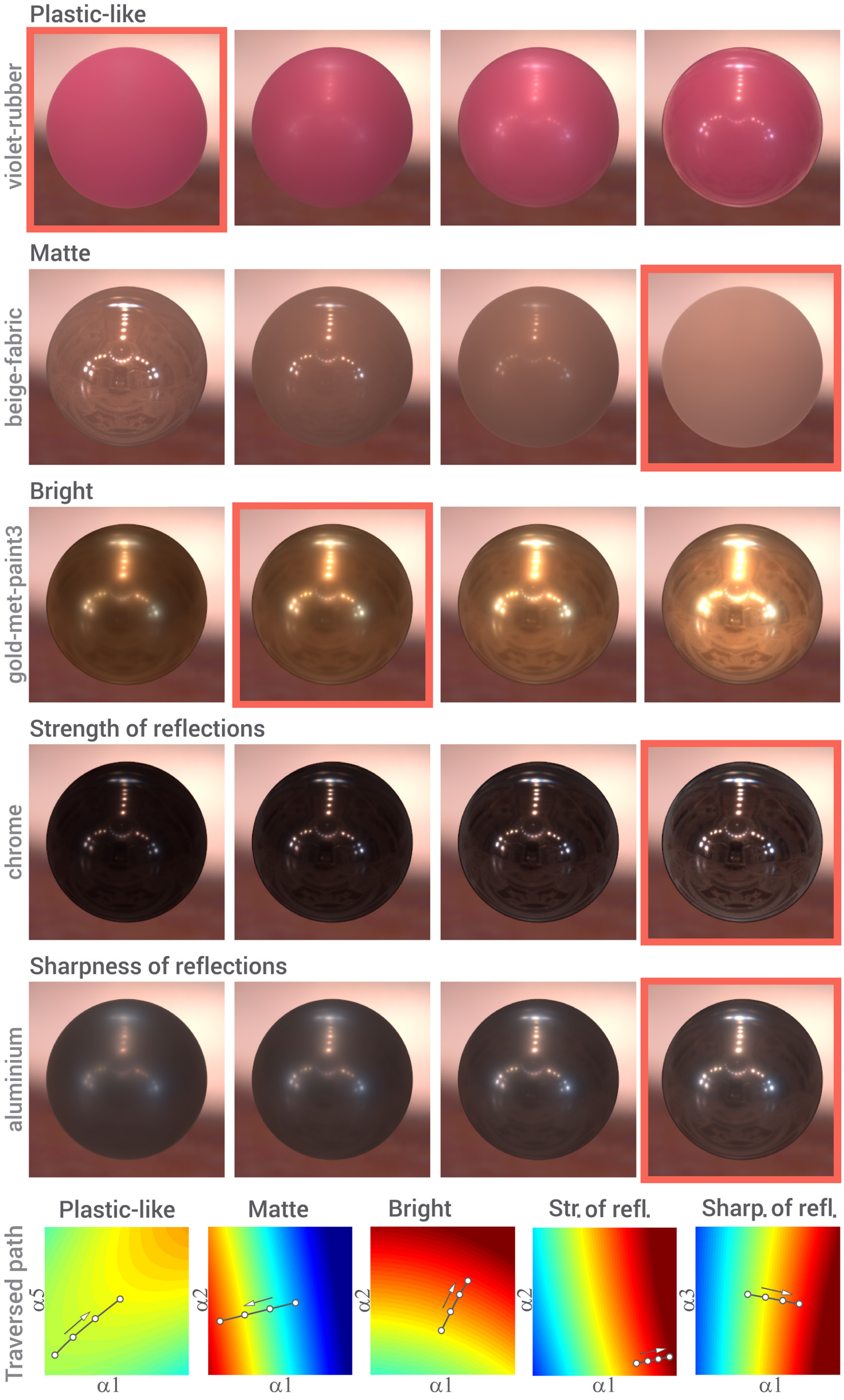


Figure 4.11: Editing BRDFs by varying the values of our attributes. Each row shows an original BRDF from the MERL database (marked in orange), and the results when linearly increasing or decreasing the value of the given attribute. The last row shows the path followed when traversing our 5D space (most representative 2D slice).

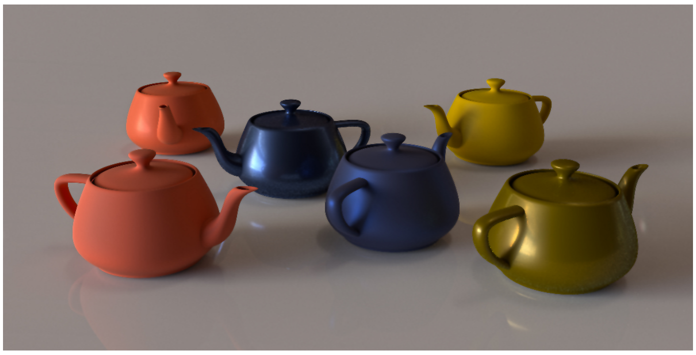


Figure 4.12: Scene rendered with measured materials from the MERL database and edited BRDFs using our framework. From left to right, we modify MERL's *pink-plastic* BRDF (foreground) by increasing the *ceramic-like* attribute (background); we modify MERL's *blue-metallic-paint2* BRDF (background) by increasing the *rough* attribute (foreground); we modify MERL's *yellow-paint* BRDF (background) by increasing the *metallic-like* attribute (foreground).

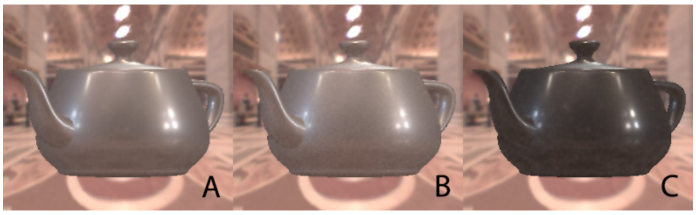


Figure 4.13: Our functionals allow to define similarity metrics based on individual appearance attributes. While A-B and A-C are very similar according to RMSE, A-B exhibit a much lower distance in terms of brightness than A-C.

GUIDANCE FOR GAMUT MAPPING Gamut mapping is a classic problem in appearance modeling: The goal is to ensure a good correspondence between the original model and its reproduction, overcoming the limitations of the output medium (e.g., 2D/3D printing). Since our functionals allow to obtain different BRDFs with the same perceived apperance in terms of a given attribute, we can use them to guide out-of-gamut cases back into the set of representable appearances, while ensuring that the desired *attribute* is not changed. This is shown in Figure 4.15, for two pairs of BRDFs along isocontours of the functional $\varphi(\boldsymbol{\alpha})$, and two different attributes. Note that two BRDFs located along an isocontour of an attribute should keep the same appearance in terms of that attribute, but may have different reflectance properties overall. Our functionals therefore expand the range of gamut mapping strategies beyond classic approaches.

4.8 Validation

PREDICTABILITY When a user edits a BRDF by tuning our perceptually-based attributes, she would need to know what to predict from each adjustment. Our functionals allow this (as we have shown in Figure 4.10), facilitating this desired predictability. Nevertheless, here we set out to further validate this, by verifying whether our functionals can really predict the magnitude of an attribute that is perceived by users. We design a user study, following the same procedure as in Exp. 2 (Sec. 4.4). We chose 44 BRDFs, both measured and edited, and asked 60 new participants to rate our fourteen perceptual attributes for each one. Each participant had to rate ten BRDFs. To analyze the results, we first average the ratings for each attribute and BRDF. Then we compute, for each attribute, the MSE across BRDFs between the subjective ratings and the values predicted by our functionals. We obtain a very accurate match, with MSE values below 0.02 for most of the attributes, and not getting higher than 0.03 for any of them. Tbl. 4.1 shows the exact values for the

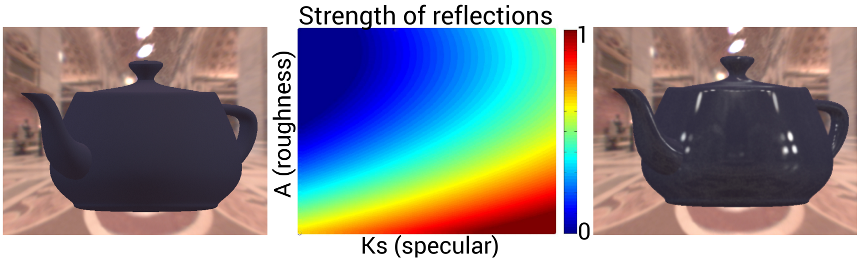


Figure 4.14: Editing the blue-fabric BRDF from MERL's database, fitted to a microfacet model (Beckmann distribution). We learn a novel functional that hides the complexity of the interactions of the different parameters in the original representation, and use it to easily alter a particular attribute. *From left to right:* original BRDF fitted to the microfacet model, learned functional for *strength of reflections*, and edited BRDF (increasing *strength of reflections*).

fourteen attributes. This close match clearly indicates that our functionals are excellent predictors of perceived appearance.

Table 4.1: MSE between the subjective ratings and the values predicted by our functionals, for each attribute, showing how our functionals are excellent predictors of perceived appearance. Error is in the range [0..1].

Attribute	**MSE**	**Attribute**	**MSE**
Plastic-like	0.0062	Matte	0.0125
Rubber-like	0.0127	Glossy	0.0178
Metallic-like	0.0127	Bright	0.0195
Fabric-like	0.0161	Rough	0.0170
Ceramic-like	0.0133	Strength of refl.	0.0134
Soft	0.0224	Sharpness of refl.	0.0214
Hard	0.0271	Tint of reflections	0.0149

PROOF OF CONCEPT WITH NOVICE USERS To assess the practicality of our control space for novice users, we designed an informal user study involving appearance editing. The task is to be carried out using both our functionals and the commercial software 3ds Max from Autodesk. We created a prototype implementation of our approach as a plugin for BRDF Explorer[11]. Users were shown images of two spheres, rendered with an initial and a final BRDF. Equivalent pairs were created with each software (our BRDF plugin and 3ds Max) to guarantee fairness, since an exact appearance match is difficult to achieve across platforms, and not a requisite for the test. The pairs were chosen so that they covered a wide range of appearances, showing complex and significant changes between the two (shown in Appendix B). Users were then asked to create spheres with an intermediate appearance, using both platforms in succession. Editing with both tools produced similar results in terms of appearance, but much faster with our prototype, as we show in Tbl. 4.2. The resulting BRDFs appear in Appendix B.

The outcome of this user study suggests that novice users find it challenging to convey a particular appearance, and that our editor can be useful in such cases. In conclusion, while commercial packages like 3ds Max offer a very good degree of control with many advanced features, for novice users our system offers a more intuitive and easy-to-use control. Both approaches are thus complementary; our system could be integrated as a plugin to these more sophisticated commercial packages, extending the range of tools available for predictable appearance changes.

[11] http://www.disneyanimation.com/technology/brdf

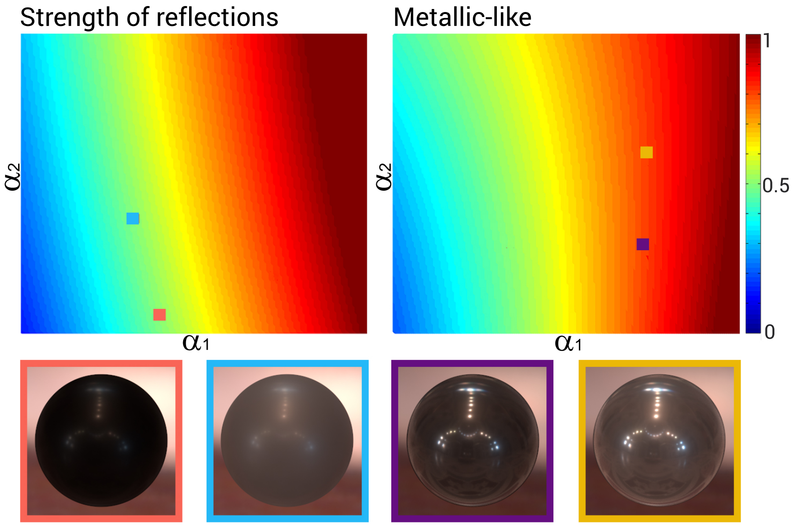


Figure 4.15: 2D slices on the $\mathbf{s}_1$-$\mathbf{s}_2$ plane for two attributes: *strength of reflections* and *metallic-like*. On each slice, we render the BRDFs corresponding to two points along an isocontour of the respective attribute. Two BRDFs located along an isocontour of an attribute should have the same appearance in terms of that attribute, but may have different reflectance properties overall. *Left:* the strength of the reflections is kept the same, despite varying roughness and diffuse properties. *Right:* the metallic quality of both BRDFs is very similar, despite having different strength and sharpness of reflections.

Table 4.2: Editing times (in seconds) with the commercial tool 3ds Max and with our prototype for each of the tasks. Note that the table includes the times actually spent editing (i.e., rendering times employed on intermediate visualization of results are subtracted from the total measured times).

			Time (in seconds)	
			3ds Max	Our prototype
Task 1	*Pair #1*	User A	41	20
		User B	194	9
		User C	94	35
	Pair #2	User A	95	40
		User B	108	66
		User C	39	35
	Pair #3	User A	76	38
		User B	80	43
		User C	91	29

4.9 Building and navigating a soft PC-space

As Nielsen et al.[236] already noticed, sometimes artifacts appear when representing very diffuse materials with the derived PC-space. These artifacts are caused by the bias towards specular materials in the MERL database, this makes the PCA decomposition less accurate for diffuse materials. A simple way of addressing this issue was proposed by Nielsen et al., and consists on splitting the database into *soft* and *specular* materials, and using the respective PC-space derived for representing a material. However, the task of switching representations when needed while traversing our space is not straight forward. We describe in this section how to build a *soft* PC-space and traverse through our space with both PCA representations (soft and specular).

4.9.1 *Training a soft PC-space*

We train our *soft* PC-space as described in Section 4.3.1, but with a reduced subset of BRDFs. The range of materials that the PC-space can accurately represent depends heavily on the BRDFs used to perform the PCA decomposition. In this case we want to represent correctly diffuse materials and avoid the bias towards specular BRDFs. Thus, we choose a subset of 40 BRDFs from the MERL database by manually selecting diffuse materials based on their appearance and the shape of their specular reflections. Figure 4.16 shows a diffuse BRDF represented both with the original PC-space trained using the full MERL database (*full* PC-space), and the *soft* PC-space trained using a subset of diffuse BRDFs. Finally, we train new funcionals that relate our perceptual attributes to the first 5 components of this new representation.

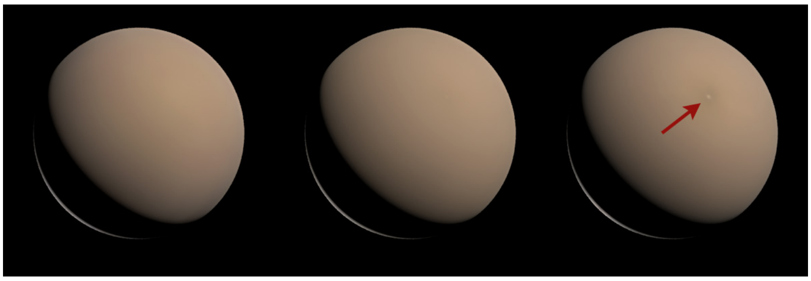


Figure 4.16: The original MERL BRDF white-fabric (left), compared with its representation with our soft PC-space(center) and the PC-space trained with the entire MERL database. The artifacts of the diffuse material disappear in the new soft PC-space.

4.9.2 *Navigating between representations*

In order to effectively navigate our space, we first need to know which representation suits best a given BRDF. In order to do so, we define a measure of specularity that will allow us to select the appropriate representation for every BRDF in our space. Second, we need a way to match the same BRDF on both representations, and to navigate between them, i.e., given the full PC coefficients of a BRDF, find its soft PC coefficients without reconstructing the entire measured BRDF.

MEASURING MATERIAL SPECULARITY The work of Nielsen [236] shows that BRDFs can be acquired with few measurements in certain specific angles. This means that the characteristic behavior of a material can be extracted from these directions. Our hypothesis is that the specularity information of the material can be obtained by just looking at the values stored on these directions. We propose as specularity measure an average of the RGB values for the most significative direction given by Nielsen on his work, as this is a simple yet effective method to acquire information about the specularity of a material. We then train a Support Vector Machine (SVM) with the specularity measures from the BRDFs in the MERL database. This allows us to binary classify, and thus choose the correct representation, for every new BRDF generated within our space.

SWITCHING REPRESENTATIONS Once we know which representation to use, we need a technique to navigate between representations without reconstructing the whole measured BRDF, as this would slow down the process of traversing the space. We again make use of RBF networks as described in Section 4.5.1, but this time we train them to learn the correspondences between the first 5 components of the *soft* and *full* PC-spaces. We perform the cross-validation to minimize the average error and find the optimal parameters ($N_c = 100$ and $\beta = 100$).

4.10 Discussion and conclusion

We have presented an intuitive control space for material appearance, which allows for artistic exploration of plausible material appearances based on perceptually-meaningful attributes. We have significantly extended the original MERL database to include 400 BRDFs, both captured and synthesized. We have derived novel functionals connecting principal components of the BRDF to a high-level characterization of material appearance, inferred from 56,000 answers collected in a large-scale study with 400 participants. This characterization is made up of our appearance attributes, which are intuitive, descriptive, and discriminative with respect to many different reflectance properties, as we have shown. We have further analyzed the resulting appearance space, which has yielded insights on material perception, and proposed a number of example applications that can benefit from our approach. Our framework aligns changes of the attribute values with predictable appearance changes. Similar to related works [220, 296], our attributes are not orthogonal; this is to be expected, and we have shown that indeed some appearance characteristics are highly correlated.

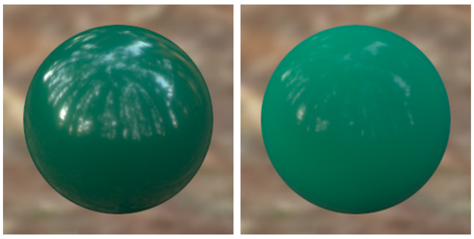


Figure 4.17: Limitation example. Since our dataset does not contain a large number of fabric-like samples, editing towards that goal from a very different initial BRDF may fail. In this case, our system cannot remove the specularities present in the original BRDF.

There are many opportunities for interesting future work. First, we do not claim to have found a complete, universal list of perceptual attributes defining appearance. This is an open problem, for which no established methodology exists. In fact, a key advantage of our flexible methodology is that it allows to define *custom* attributes, which may adapt better to a particular user or context, while avoiding mixed nomenclatures. Moreover, it can also be used on different databases. Second, it would be interesting to expand our approach to more materials; despite the fact that our extended MERL dataset provides a reasonably uniform coverage over a very wide range of isotropic appearances, some perceptual attributes can be under-represented. This translates into less user ratings, which may lead to less reliable functionals in some regions of our 5D space (see Figure 4.17). Similarly, for some BRDF clusters and specific attributes we find that the variance in scores is relatively high (e.g., *ceramic-like* for the *metallic* cluster in Figure 4.7). This seems to indicate that subjects do not agree on how *ceramic-like* the BRDFs in that cluster are. Our functionals will thus be less reliable in that case, as a consequence of people not agreeing on perceptual appearance. Large variance in scores for an attribute in a cluster, however, can have different causes: Some attributes have a large variance in scores, but a high agreement (e.g., *rough* for the *metallic* cluster in Figure 4.7), seemingly indicating that the large variance in scores comes from the fact that that particular attribute can exhibit a range of different values within the cluster; in the case of the metallic cluster, BRDFs show a wide range of roughness. Our data, however, is not enough to state strong conclusions in this regard. Further, our system does not currently handle some complex appearance behavior such as color changes, grazing angle effects, or hazy gloss. These are undersampled in our dataset, and remain as future work, deserving further investigation. Last, despite the many insights gained in this work, a full exploration of our space for material appearance still remains an exciting open task.

We hope that our work can inspire additional research, in addition to the four applications we have shown. For instance, it could help to better understand the underlying perceptual aspects of analytic models, or to find a perceptual scaling for their parameters (Figure 4.14 shows a proof of concept mapping between perceptual attributes and analytic BRDFs). It could also help to examine the representational space of existing models, to design computational fabrication techniques to achieve a desired appearance, or even to develop efficient BRDF sampling strategies.

5. Attribte-preserving gamut mapping of measured BRDFS

About this chapter

The work presented in this chapter has been presented at *Eurographics Symposium on Rendering* (EGSR) 2017 and published in *Computer Graphcis Forum*. This is a follow-up of the work presented in the previous chapter 4. In this work we focus on a particular application of our perceptually-based space, gamut mapping. We propose a two-step mapping algorithm; my contribution lies mainly in the first step of the algorithm, which is carried out based on the perceptual traits derived in our previous work and takes into account only achromatic reflectance, whereas Tiancheng Sun, during his internship at *Universidad de Zaragoza*, developed the second step of the algorithm, which takes place in image space and also takes into account chromaticity. Previous stages of this work were presented as a poster at SIGGRAPH 2017. This poster was awarded with the 1^{st} place at the ACM SIGGRAPH Student Research Competition (undergraduate category), and later it was also awarded with the 1^{st} place at the ACM Student Research Competition Grand Final.

T. Sun, A. Serrano, D. Gutierrez, and B. Masia.
Attribute-preserving gamut mapping of measured BRDFs.
Computer Graphics Forum, Vol. 36(4), 2017

T. Sun, A. Serrano, D. Gutierrez, and Belen Masia.
Attribute-preserving gamut mapping of measured BRDFs.
In ACM SIGGRAPH (posters), 2017.

5.1 Introduction

Real-world materials present a wide variety of appearances, commonly described in computer graphics with the bidirectional reflectance distribution function (BRDF). Current printers, on the other hand, have a predefined set of only a few inks, which in turn defines the printer's gamut. As a consequence of this limitation, many materials cannot be exactly reproduced by the printer. Finding the best approximation of the input BRDF that falls within the printer's gamut is the problem known as BRDF gamut mapping.

Gamut mapping is an extremely underconstrained problem without a unique solution, for which several methods have been proposed. The usual approach is to try to find the BRDF which is *most similar* to the target BRDF, while lying within the available gamut. In this work, we add a novel approach to the state of the art: Our gamut mapping technique allows the user to set a certain perceptual attribute (or attributes) that needs to be preserved as much as possible while mapping the BRDF into the available gamut. For instance, the user may specify explicitly the preservation of the *strength of reflections*, or the *metallic* appearance of the material.

To achieve our goal, we propose a two-step gamut mapping technique: In the first step, we leverage recent works on material acquisition [236] and editing [289]. In these works, Nielsen et al. first built a five-dimensional PC space which serves as a basis for representing each BRDF; then, Serrano et al. learnt functionals mapping the space of principal components to higher level perceptual attributes defined for achromatic reflectance; these functionals define an intuitive control space for appearance. We use these mappings in PC space to follow the path that brings the luminance channel L (L in Lab space) into gamut in PC space, while preserving the desired attribute (see Figure 5.1). However, adding the *ab* color coordinates to the remapped *L* leads to a BRDF that will most likely still be out of gamut (Figure 5.1). We thus complete the gamut mapping process with our second step, which consists of an image-based optimization, inspired by other recent works [233, 254].

We validate our technique on the BRDFs from the MERL database [220], using a gamut formed by real measured inks [221]. We show that, for a majority of BRDFs in the database, performing this first step we propose leads to a better final result than that obtained by previous works. Given that a definitive metric for BRDF similarity does not exist, in addition to computing a BRDF metric, we validate our results by means of a user study. Although it is not a requirement of the method, our proposed method allows certain interactivity, since the user can choose which attribute(s) to preserve. In the following, when performing comparisons to the state-of-the-art automatic method, we always fix the same two attributes, in order to provide a fair comparison to it. However, we also show results preserving other attributes, showing this additional feature of our method.

5.2 Related work

In this section we focus on perceptual spaces for BRDFs and gamut mapping works; we refer the reader to more general recent surveys on perception and graphics [224] and BRDF representation [121] for a broader view.

5.2.1 *Perceptual spaces for BRDFs*

Predicting the appearance of a given BRDF can be complicated: not only do the shape of the object and the lighting environment affect the perceived appearance of the material [102, 326], but directly changing the parameters of analytical models usually results in unintuitive and perceptually non-uniform changes as well. Observing this, Pellacini *et al.* derived a perceptually uniform parameter space for the Ward BRDF model [252]. Ngan *et al.* proposed an image-driven BRDF metric, which allows users to linearly modify the material's appearance using several analytical models. A perceptual space for gloss was later proposed based on real material measurements [343]. Havran *et al.* pushed forward perceptually-motivated BRDFs by further considering the lighting and viewing directions in a single image [130]. Matusik and colleagues presented the MERL database of measured BRDFs [220], proposed two techniques for dimensionality reduction, and had a single user classify whether a given BRDF possesses a series of *perceptual traits* or not, which in turn allowed them to modify a BRDF's appearance intuitively. Recently, Serrano *et al.* [289] extended the MERL dataset and presented a series of functionals connecting such data with the perceptual ratings obtained from large-scale user studies. The derivation of these perceptual spaces can be used for gamut mapping, in particular for the establishment of distances among BRDFs which is required for gamut mapping. In this work, given the underconstrained nature of gamut mapping, we take the approach of preserving certain aspects of appearance, and in particular those defined by one or more perceptual attributes. Therefore, we use the functionals derived by Serrano *et al.* [289] to map the original BRDF to the nearest material inside the gamut that preserves certain perceptual attributes.

5.2.2 *Gamut mapping for BRDFs*

Several approaches have been proposed to reproduce specific material appearances. Weyrich *et al.* [341] used a grid of tilted small facets to achieve custom reflectances; Matusik *et al.* [221] used halftoning to print a certain appearance on paper using a regular printer. There are also works focusing on reproducing subsurface scattering [85, 129]. A model which can cover both isotropic and anisotropic appearances was proposed by Lan *et al.* [178], which combines the micro-facet model with BRDF printing. Closer to our approach, other works focus on finding the closest material inside the valid gamut of a printer. The metric on half-angle curves of the materials was used to resolve the best components of the inks in the work of Matusik et al. [221], while Lan *et al.* [178] calculated the L2 norm on the BRDF hemisphere for data fitting. Pereira and Rusinkiewicz improved this procedure by minimizing the difference between rendered images of the original and the final materials [254]; one slice of the BRDF image was calculated analytically for optimization. Similar to this work, we also perform an image-based optimization; however, we do this as a second step in our gamut mapping algorithm, after finding an intermediate BRDF that better preserves the desired perceptual attributes of the original BRDF.

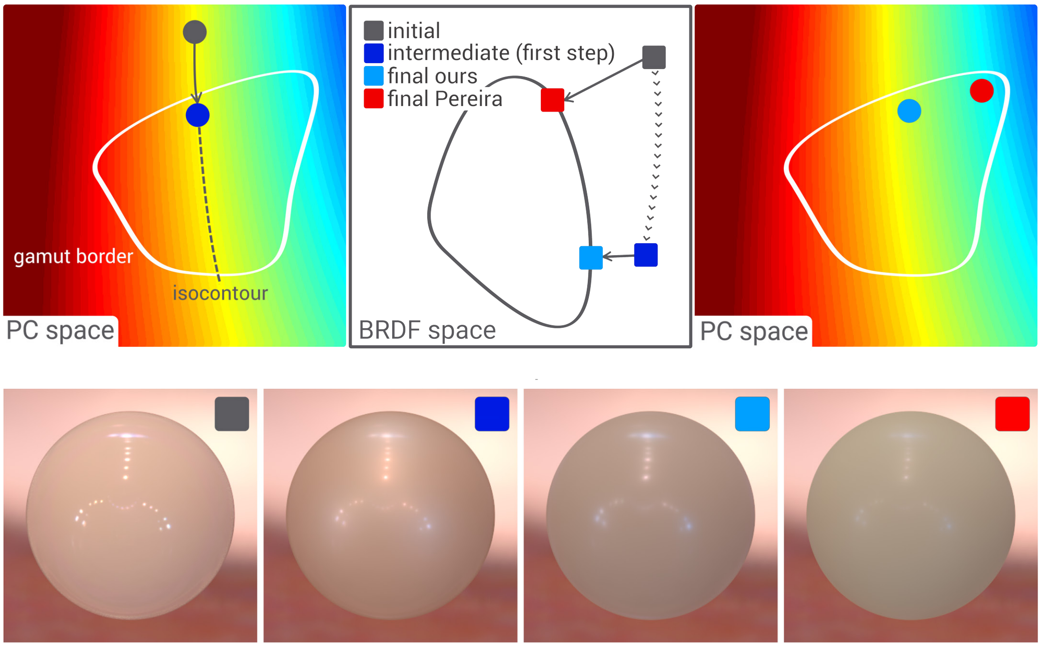


Figure 5.1: Overview of our two-step gamut mapping method. **Top-left:** A two-dimensional projection of the five-dimensional PC space. The white line represents the border of the gamut, and same color-coded isocontours indicate the same value of a given perceptual attribute (following [289]). Working on achromatic reflectance, we first push the original (target) BRDF (gray) into gamut (intermediate BRDF, dark blue) in such PC space. **Top-middle:** Back in the original BRDF space, the intermediate BRDF is not guaranteed to be in gamut (the dotted line represents the previous move in PC space); we therefore apply an image-based optimization to bring it into gamut (final BRDF, light blue). The red dot represents the result of applying a single step based on image optimization [254]. **Top-right:** For comparison purposes, we move back to PC space to show the final BRDFs with both methods; ours (light blue) lies much closer to the intended attribute value than the single-step method (red). **Bottom:** Real results with the *alumina oxide* BRDF from the MERL database. From left to right: original (out of gamut); our intermediate BRDF (still out of gamut); our final result; single-step image-based optimization [254]. Our result preserves highlights better, while exhibiting less color shift.

5.3 Attribute-preserving gamut mapping

Our goal is to take an out-of-gamut BRDF and bring it into a representable gamut, defined for instance by the individual color inks of a printer, while preserving a given perceptual attribute, such as its *brightness*. However, mapping perceptual attributes of a material to its underlying BRDF representation is not an obvious task. Recently, Serrano et al. [289] gathered a large number of subjective ratings on the high-level perceptual attributes that best described their extended MERL BRDF dataset. Using these ratings, the authors then built and trained radial basis functions networks, which are used as functionals mapping the perceptual attributes to a five-dimensional log-mapped PCA-based representation of the BRDFs, proposed by Nielsen and colleagues [236].

These functionals, derived for achromatic reflectance only, are good predictors of the perceived attributes of appearance. We leverage this work to develop our two-step gamut mapping algorithm.

Figure 5.1 qualitatively presents an overview of our method. First (Figure 5.1, top left), working in PC space, we follow the isocontour of a given functional to bring into gamut (in PC space) the initial BRDF; these isocontours represent the same value of a given attribute (please refer to [289] for details). For visualization purposes we show a 2D slice of the original 5D space. This implies fixing values for the other three dimensions to select which slice to plot, and projecting the points corresponding to the BRDFs onto the slice. As a result, points in a particular projection may appear slightly inside the gamut, even though they do lie on the border in 5D space. Note that in this space we only work with L values; color will be handled in a second step. This yields an intermediate BRDF which, although it preserves the desired perceptual attribute, cannot be guaranteed to fall within the gamut defined by the inks in the original BRDF space. In our second step, we bring the intermediate BRDF into gamut using an image-based optimization (Figure 5.1, top center). Figure 5.1, top right, shows the final BRDF using our method, compared to a single-step, image-optimization method (such as Pereira's state-of-the-art algorithm [254]) in PC space. It can be seen how our result better preserves the intended attribute in this space, since it is never explicitly taken into account in existing single-step methods. Figure 5.1, bottom, shows real examples produced with our method, and Pereira's [254]. We use the *alumina oxide* BRDF from the MERL database, and aim to preserve the *metallic-like* and *bright* attributes. Although obvious differences exist in both results with respect to the original BRDF, given the limited ink gamut, our method maintains stronger highlights and exhibits significantly less color shift.

Note that in our second step, we optimize both the L and the chromatic coordinates (a, b), despite the fact that L had already been modified in the previous step. This is because our gamut is defined by a series of real-world inks, in which L cannot be isolated and optimized independently of chromaticity. In other words, although decoupling achromatic reflectance from color information was convenient in PC space to follow an isocontour that would preserve the value of a certain attribute, they cannot be decoupled when handling real inks. Modifying such ink components independently would not guarantee that the final BRDF lies within the available gamut. Our first step provides, however, a better starting point for the image-space optimization step, leading to better results (see Section 2.5). In the rest of the section we describe our two steps in detail.

5.3.1 *Step 1: Luminance mapping in PC space*

We call our initial BRDF $\boldsymbol{\rho}_{\text{ini}}$. In this first step, our goal is to obtain an intermediate BRDF $\boldsymbol{\rho}_{\text{int}}$, where we bring the L channel into gamut in log-mapped PC space [236] following the path that maintains the same value v_{A} of a certain perceptual attribute A[12]. Since the functionals derived in the work of Serrano et al. are learned with respect to the coefficients for L in PC space, we follow here the same procedure. First, we apply a log-relative mapping to $\boldsymbol{\rho}_{\text{ini}}$, which enables a good distribution of the available dynamic range [222, 236, 289], and obtain the first five coefficients $\boldsymbol{\alpha}_{\text{ini}} \in \mathbb{R}^5$ of the BRDF in the PCA basis (which provide general hints about material appearance [236]). Then, we apply a function $f : \mathbb{R}^5 \to \mathbb{R}$ that maps a BRDF in the aforementioned 5D PC space to its attribute value [289]: $v_{\text{A}} = f(\boldsymbol{\alpha}_{\text{ini}})$. This function f is a radial basis function network (RBFN) based on Gaussian functions such that:

$$\varphi(\boldsymbol{\alpha}) = \sum_{\text{i}=1}^{N_{\text{c}}} \theta_{\text{i}} \exp^{-\beta \|\boldsymbol{\alpha} - \mathbf{c}_{\text{i}}\|^2} \tag{5.1}$$

where β controls the smoothness of the Gaussian functions, N_{c} is the number of neurons in the network, c_{i} are the centers of such neurons, and θ_{i} are the weights of each neuron. RBFNs are a particular type of artificial neural networks called static neural networks, where the outputs are linear combinations of radial basis functions, and the neurons correspond to cluster centers.

[12] For the sake of clarity, we explain our method for the simpler case of fixing just one attribute.

For our gamut mapping, we formulate the objective function to maintain the initial attribute value $v_{\mathrm{A}} = f(\boldsymbol{\alpha}_{\mathrm{ini}})$, so that the optimization moves along the corresponding isocontour of f as much as possible. Our objective function is therefore:

$$g(\boldsymbol{\alpha}) = \varphi(\boldsymbol{\alpha}) - \varphi(\boldsymbol{\alpha}_{\mathbf{ini}}) \tag{5.2}$$

Further, we need to ensure that the resulting BRDF is inside the gamut in PC space, which we formulate as a hard constraint. We define the gamut as the set of possible convex combinations of the inks, expressed in our formulation as the convex hull $Conv(\alpha_{\mathrm{inks}})$ limited by the ink coefficients in the 5D PC space. The resulting optimization problem becomes:

$$\min_{\alpha} \|g(\boldsymbol{\alpha})\|^2 \quad \text{s.t.} \quad \alpha \in Conv(\alpha_{\mathrm{inks}}) \tag{5.3}$$

In order to solve this optimization we use sequential quadratic programming (SQP) [344] as implemented in the *fmincon* function in *MATLAB*. In this way, we obtain the new PC coefficients $\boldsymbol{\alpha}$ defining our intermediate BRDF $\boldsymbol{\rho}_{\mathrm{int}}$, which lies inside the inks gamut in PC space, while keeping the value v_{A} of the desired attribute A from the initial BRDF $\boldsymbol{\rho}_{\mathrm{ini}}$.

If instead of one we want to preserve multiple attributes, we employ a linear combination of them in the objective function. In principle, we weight all attributes equally in this linear combination, but alternative weighting according to the user's intent is also possible.

5.3.2 *Step 2: Image-space optimization*

After the first step we have an intermediate BRDF $\boldsymbol{\rho}_{\mathrm{int}}$ which is not necessarily inside the gamut defined by the inks, since we have optimized for achromatic reflectance L only in log-mapped PC space. Let us consider a gamut defined by a set of N inks; any reproducible BRDF lies inside the convex hull formed by the inks' BRDFs $\mathbf{P}_{\mathrm{inks}} = [\, \boldsymbol{\rho}_1 \quad \boldsymbol{\rho}_2 \quad \cdots \quad \boldsymbol{\rho}_{\mathrm{N}} \,]$ in BRDF space. Our goal is to find a BRDF $\boldsymbol{\rho}_{\mathrm{fin}}$ that lies within this convex hull, and is thus a convex combination of $\mathbf{P}_{\mathrm{inks}}$ such that:

$$\boldsymbol{\rho}_{\mathrm{fin}} = [\, \boldsymbol{\rho}_1 \quad \boldsymbol{\rho}_2 \quad \cdots \quad \boldsymbol{\rho}_{\mathrm{N}} \,] \cdot \mathbf{w}, \tag{5.4}$$

where $\mathbf{w} \in \mathbb{R}^N$ is the vector formed by the coefficients of the convex combination.

When trying to minimize the distance between the intermediate BRDF $\rho\rho_{\mathrm{int}}$ and the reproducible one $\rho\rho_{\mathrm{fin}}$, we can in principle work in BRDF space, or in image space. Working in BRDF space with measured BRDFs is very costly, due to the large size of the data; furthermore, similarity of the raw BRDF data does not imply visual similarity [104]. Therefore, we will instead minimize this distance in image space, as has been done in the past [254].

Given the incident light, geometry, and BRDF, the per-pixel image formation model is described by the rendering equation:

$$L_{\mathrm{out}} = L_{\mathrm{emit}} + \int_{\Omega} \rho(\omega_{\mathrm{i}}, \omega_{\mathrm{o}})\, L_{\mathrm{in}} \cdot (\omega_{\mathrm{i}} \cdot \mathbf{n})\, \mathrm{d}\,\omega_{\mathrm{i}}. \tag{5.5}$$

where L_{out} is the outgoing radiance in direction ω_{o}, L_{in} is the incident light in direction ω_{i}, $\mathbf{n}$ is the surface normal, and Ω is the hemisphere defining the integral domain; we consider additional light emitted by each surface point L_{emit} to be zero. Given inter-reflections and indirect lighting, this equation defines a non-linear process. However, if we consider a single convex object (a purely opaque sphere, commonly used to visualize BRDFs) light interacts only once on its surface before reaching the camera. Thus, the rendered image and the BRDF are linearly related as:

$$\mathbf{I} = \mathbf{R} \cdot \rho, \tag{5.6}$$

where $\mathbf{R}$ is a matrix defining the linear mapping. Using Equation 5.6, we can change Equation 5.4 into:

$$\mathbf{I} = [\, \mathbf{I}_1 \quad \mathbf{I}_2 \quad \cdots \quad \mathbf{I}_{\mathrm{N}} \,] \cdot \mathbf{w}, \tag{5.7}$$

where $\mathbf{I}_i$ are the rendered images for each ink. Note that this sidesteps the need to explicitly compute $\mathbf{R}$; moreover, these rendered images allow to establish visual similarity better than raw BRDF data. We can now use these rendered images to obtain the optimal coefficients $\mathbf{w}_{\text{opt}}$:

$$\mathbf{w}_{\text{opt}} = \underset{\mathbf{w}}{\operatorname{argmin}}\ d(\mathbf{I}_{\text{int}}, \begin{bmatrix} \mathbf{I}_1 & \mathbf{I}_2 & \cdots & \mathbf{I}_N \end{bmatrix} \cdot \mathbf{w}) \\ \text{s.t.}\ \sum \mathbf{w} = 1,\ 0 \leq \mathbf{w} \leq 1.$$

where $\mathbf{I}_{\text{int}}$ is the image obtained with the BRDF computed in the first step. This distance d is thus computed in image space, and could be done in RGB or in Lab color space. We choose d to be the L_2 norm under Lab color space, since it better preserves the color of the original BRDF; we show an example of this in Figure 5.2.

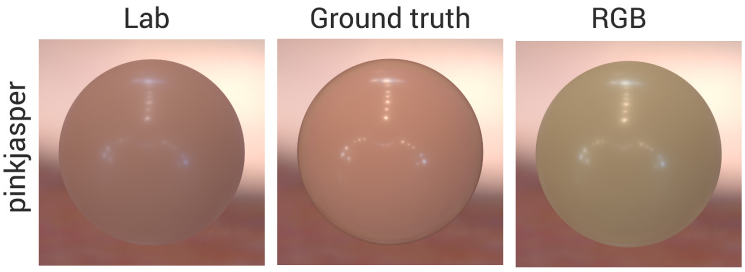


Figure 5.2: Gamut mapped results for the *pinkjasper* MERL BRDF (middle) optimizing in Lab (left) and RGB (right). Image-based optimization in Lab space better preserves chromaticity. The set of inks that define the gamut can be seen in Figure 5.3.

5.4 Results

When computing our results, we use the BRDF gamut from Matusik et al. [221]. This gamut contains 57 BRDFs, which are measured from real world inks. Since the gamut is designed to reproduce a wide range of material appearances, most of the inks are specular and metallic, which are not found in standard printers (the inks are shown in Figure 5.3). Our images of the inks and initial BRDF used in the optimization are rendered with the *St. Peter's Basilica* environment map, while for the results used in our user study (Section 5.4.2) we use the *Eucalyptus Grove* map[13], given that these illuminations facilitate material perception [102], and in order to employ different illuminations for validation and optimization. In all results shown in this section, except when noted otherwise, we are performing gamut mapping preserving the *metallic-like* and the *bright* attributes in the first step. All the images have been equally tonemapped adjusting exposure and gamma.

Figure 5.4 shows the influence of our first step (luminance optimization in PC space) in the final results, as opposed to using only the image-based optimization of the second step. Our final result (two steps) is much closer in appearance to the target, out-of-gamut BRDF (*twolayergold*) than the result of a single-step image-based optimization (i.e., without the first attribute-preserving step). The effect of the first step (although in this case compared to the image-based optimization of Pereira [254]) can also be seen in Figure 5.1.

5.4.1 *BRDF gamut mapping results*

We compare our results with those from Pereira [254] on 94 homogeneous materials from the MERL database [220]. Some representative results rendered with different environment maps are shown in Figure 5.5 (please refer to the project website for the complete set). To provide an

[13] Both environment maps are from the Light Probe Image Gallery [76].

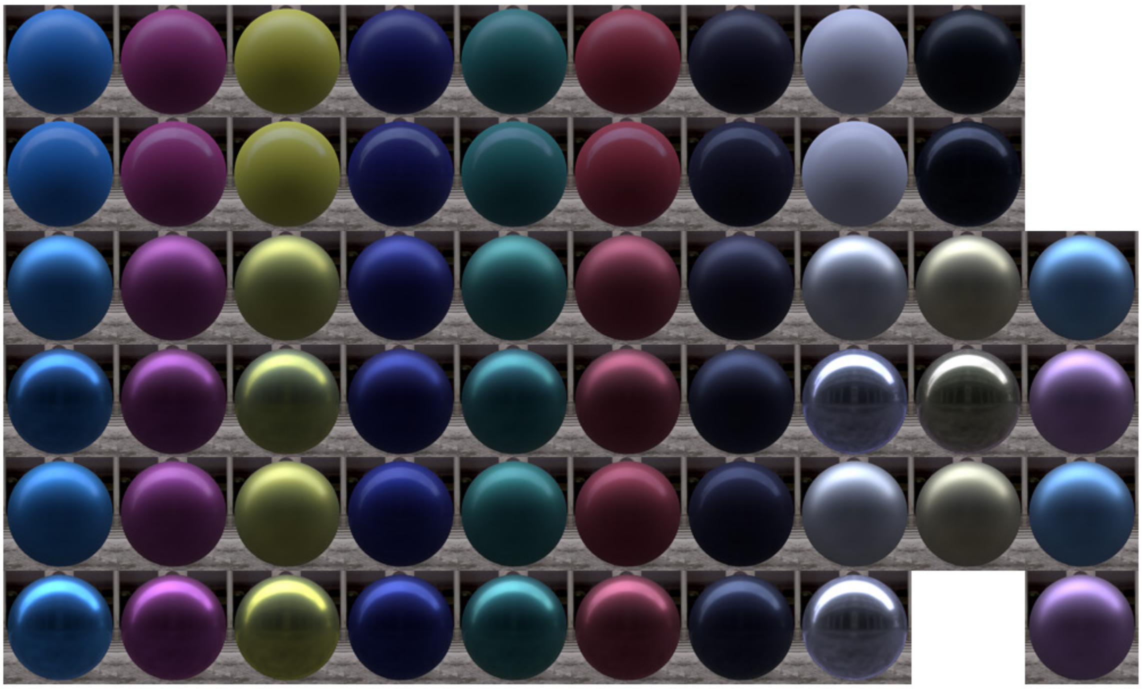


Figure 5.3: All the BRDFs from the gamut provided by [221], which are measured from real inks.

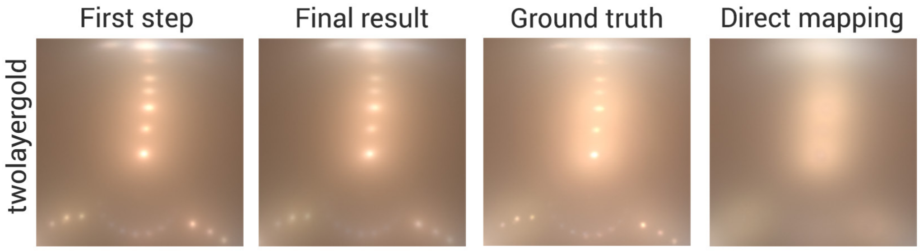


Figure 5.4: Our two-step, attribute-preserving gamut mapping, compared to a single-step optimization. Note how our method helps preserving the specular highlights and overall appearance better.

objective comparison to the state of the art [254], we use the cube root cosine weighted RMS metric, which has been reported to perform better than RMS for BRDFs [104]. This metric is described as:

$$E = \sqrt{\frac{\sum_{\text{n}}((\rho_{\text{fin}} cos\theta_{\text{i}} - \rho_{\text{ini}} cos\theta_{\text{i}})^2)^{1/3}}{n}} \qquad (5.8)$$

where θ_{i} is the the cosine of the angle between the incident light and the normal. Results of this metric are shown for all MERL BRDFs in Figure 5.6. We plot the difference between the error of both methods, sorted by increasing values, where blue indicates better results with our method (our error is lower) and red the opposite (our error is higher). Although we do not outperform the single-step method of Pereira [254] for all BRDFs in the database, we do in a majority of cases. The user study in the next subsection confirms this.

5.4.2 *User study*

Additionally, we have carried out a perceptual study to evaluate the results of our gamut mapping algorithm, with the same gamut used for the previous objective evaluation. We used a subset of

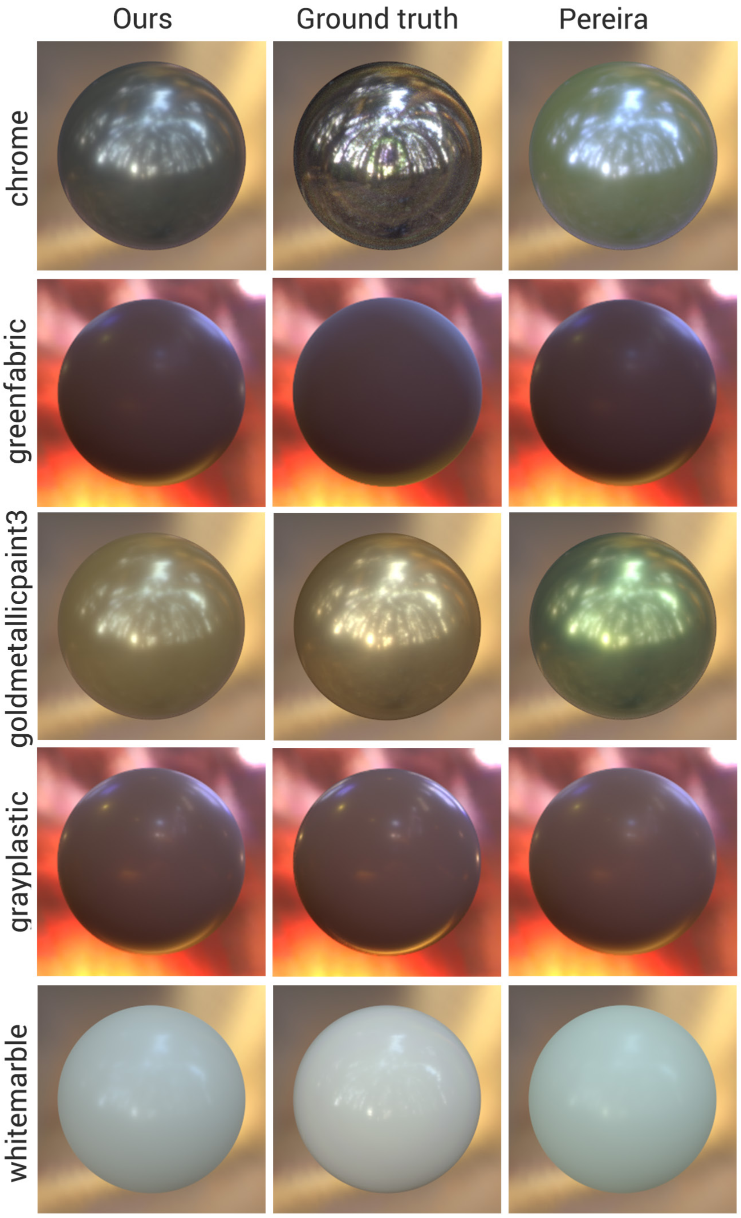


Figure 5.5: Comparison of our results and the state-of-the-art method by Pereira [254] using different illuminations (*Eucalyptus* and *Grace*). In general, our method minimizes color shifts (*chrome* and *goldmetallicpaint3*), while better preserving highlights and specular behavior (*grayplastic* and *whitemarble*). For very diffuse materials (*greenfabric*) neither method succeeds due to the specular nature of the inks used (see Figure 5.3).

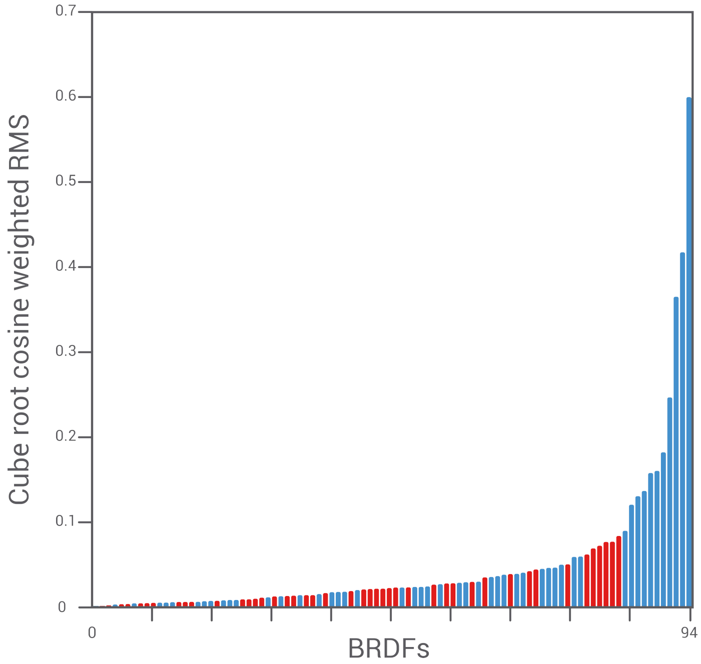


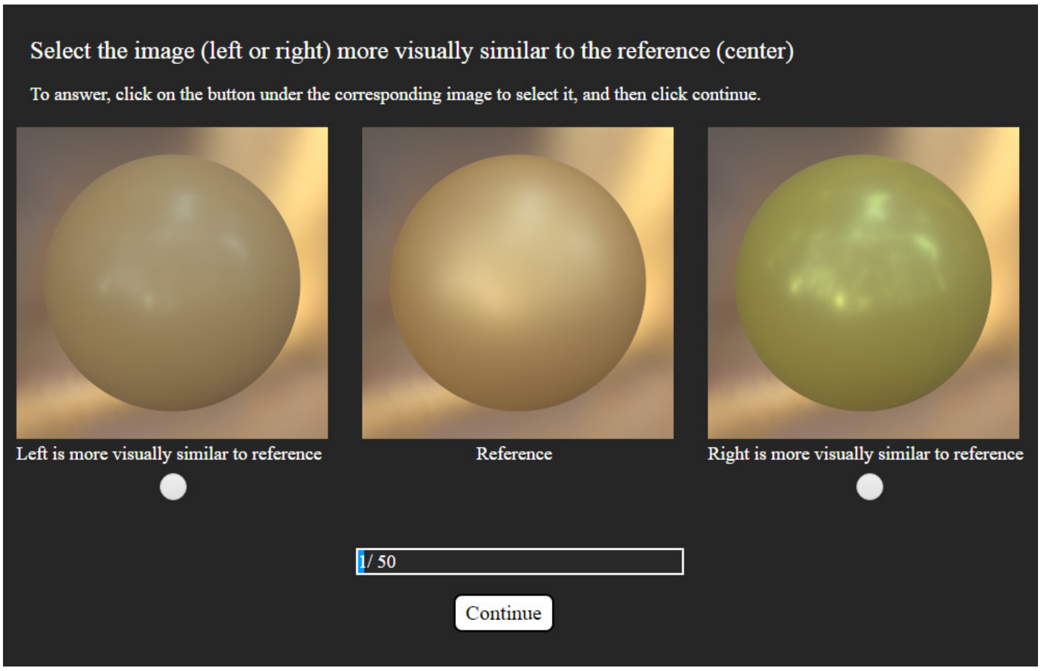
Figure 5.6: Difference between the error of the state-of-the-art image-based optimization [254] and that of our method; the error is computed as the cube root cosine weighted RMS [104]. Blue indicates better results with our method.

50 out-of-gamut BRDFs from the MERL dataset, discarding in-gamut BRDFs and BRDFs lying very far away from the gamut (see Section 5.5). Similar to previous works [102, 252, 254] we use a sphere to depict the materials. The design of our user study is based on the one reported by Pereira and Rusinkiewicz, in order to provide a fair comparison. We render them under the *Eucalyptus* environment map. In our study the user is presented with a reference image (center), and two different results (Pereira's and ours), one at each side (see Figure 5.7). The order in which the BRDFs appeared, as well as the position of each result relative to the ground truth, was randomized. Subjects were asked to select which of the two alternatives shown was more visually similar to the reference image. They were instructed that by *visually similar* we meant which of the two better represented the material appearance of the ground truth sphere.

We recruited fifteen subjects (nine male, six female). All subjects were presented with all 50 BRDFs, and the time to completion of the experiment was approximately 10 minutes. There was no time limit for making each choice, but once subjects moved forward to the next example, they were not allowed to go back. Twenty-two of the tested BRDFs showed significant differences in the results (χ^2 test, $p < 0.05$); among these, 77.6% of the time our result was chosen over the state-of-the-art method (see Figure 5.8). Agreement between users was high, with 81.3% users on average agreeing with the majority on a given choice. Overall, including the non-statistically significant BRDFs, our results were preferred in almost 62% of the results.

5.4.3 *Additional results*

In this section we present additional gamut mapping results preserving different combinations of attributes during the optimization along isocontours in our first step. In Figure 5.9 we show the outcome of using single attributes as opposed to a combination of several attributes. For the cases where the BRDF is far from the gamut, it is challenging to obtain a convincing result that matches the appearance of the original BRDF. In such cases, compromises in appearance need to be made in order to obtain a BRDF that can be represented in the gamut. When optimizing to



Select the image (left or right) more visually similar to the reference (center)

To answer, click on the button under the corresponding image to select it, and then click continue.

Left is more visually similar to reference
○
Reference
Right is more visually similar to reference
○
1/ 50
Continue

Figure 5.7: Screenshot of our user study. The reference is presented in the middle, with the two options at the sides in random order.

preserve only the *metallic-like* attribute, the resulting BRDF preserves the specular behavior better, while when optimizing for the *bright* attribute, the result matches the diffuse component better. When optimizing with the two attributes at the same time, the optimization reaches a compromise between both. In every case our algorithm presents a predictable behavior, and can be adapted to the user's needs. In Figure 5.10 we show additional results preserving in this case the attributes *rough* and *strength of reflections*. Note that this combination of attributes performs particularly well at preserving the look of the reflections, even for BRDFs which are very far away from the gamut.

5.4.4 *Material editing*

Our strategy to preserve attributes can also be applied to intuitive material editing, extending the capabilities of the recent work by Serrano et al. [289]. In particular, we allow the user to adjust the value of one attribute, while fixing a certain value for another one. Let $f^{att_1}(\alpha)$ and $f^{att_2}(\alpha)$ be the functionals in PC space related to the attribute to be changed, and the attribute to be fixed to a given value, respectively. Similar to the original work, we optimize $f^{att_1}(\alpha)$ looking for the α values that yield the desired attribute value y_{obj}. However, we now impose an additional hard constraint over the second attribute, effectively fixing its value y_{fix}:

$$\min_{\alpha} \left\| \varphi^{att_1}(\boldsymbol{\alpha}) - y_{\text{obj}} \right\|^2 \qquad \text{s.t.} \quad \varphi^{att_2}(\boldsymbol{\alpha}) = y_{\text{fix}} \tag{5.9}$$

This provides more accurate control over editing, and can be used to successfully perform edits over attributes originally coupled in Serrano's work. Figure 5.11 shows an example where we turn the original *yellowphenolic* BRDF more *metallic-like*, but making the result more or less bright by fixing different values of the *bright* attribute.

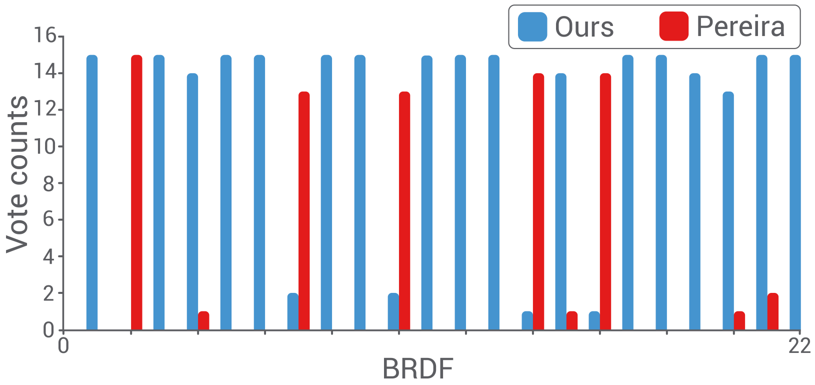


Figure 5.8: Vote counts indicating preference for the BRDFs mapped with our method (blue bars) and Pereira's method (red bars) for the BRDFs with statistically significant differences. In them, our result was preferred 17 out of 22 times with high agreement between users.

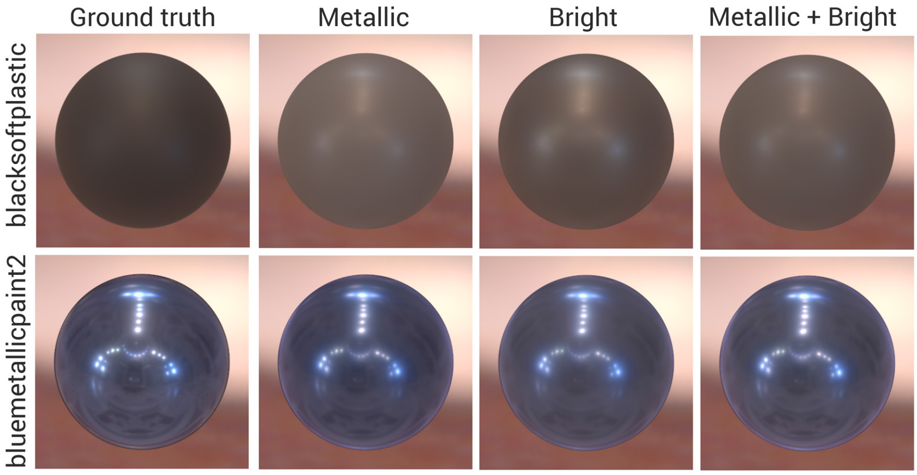


Figure 5.9: From left to right: Original BRDF and corresponding gamut mapping results computed by preserving, during the first step of our method, only the *metallic-like* attribute, only the *bright* attribute, and both attributes. Optimizing over the *metallic-like* isocontour yields more accurate reflections, while if we optimize over the *bright* isocontour the diffuse behavior is better preserved. A combination of the two attributes reaches a compromise, aiming to preserve both behaviors.

5.5 Discussion and conclusion

In this chapter, we have proposed a new two-step method for BRDF gamut mapping. In the first step we work in PC space, and use some previously proposed functionals that map this space to higher-level intuitive attributes to preserve the appearance of any of such attributes. The output of this first step, which only optimizes achromatic reflectance, is then used as input to an image-space optimization which brings the final BRDF into the ink gamut by expressing it as a convex combination of the available inks. We perform both an objective and subjective validation comparing against the state of the art. Additionally, we show how a slight modification of our framework can provide extended functionalities for intuitive material editing.

We have found out empirically that using the *metallic-like* and *bright* attributes (equally weighted) leads to good results for a large part of the MERL database (please refer to the project website[14] for results). This finding could be used to design an *automatic* gamut mapping method, since *metallic-like* tends to preserve specularities, while *bright* tends to preserve the diffuse color. Apart

[14] http://webdiis.unizar.es/~aserrano/projects/gamut_mapping

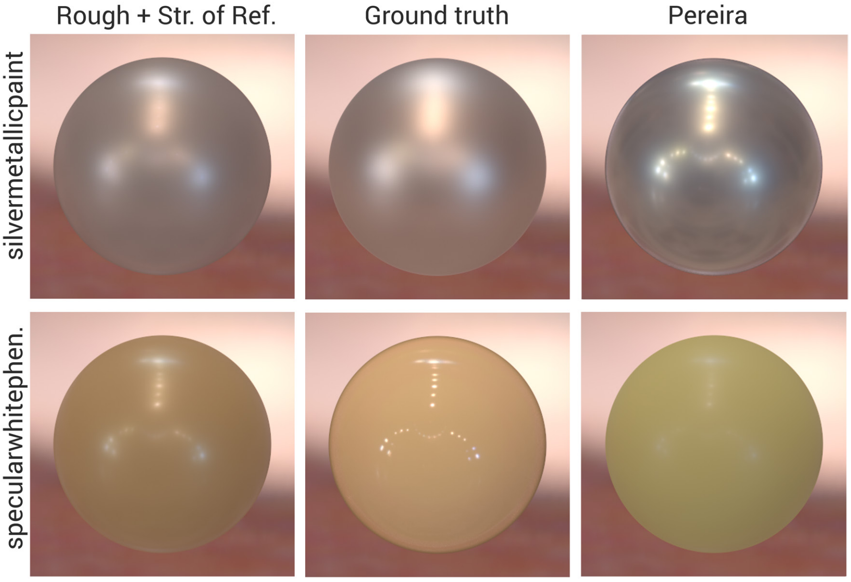


Figure 5.10: Results computed preserving both the *rough* and *strength of reflections* attributes in the first step of the method, and comparison to the state of the art [254]. This particular combination of attributes aims to better preserve the appearance of the reflections.

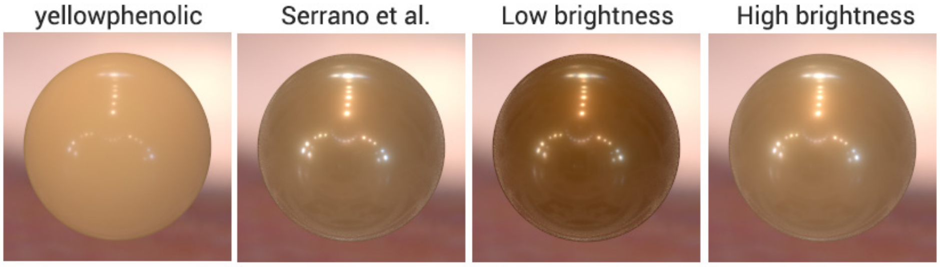


Figure 5.11: Extended capabilities for intuitive material editing, turning the original material more *metallic*. From left to right: the original BRDF; result from Serrano et al. [289] (since the *metallic-like* and *bright* attributes are coupled in their implementation, the user has no control over the final brightness of the material); our result fixing a low value for *bright*; and our result fixing a high value for *bright*.

from these two attributes, users could then tune the weight of other individual attributes, to obtain different results targeted to specific purposes.

Gamut mapping is an ill-defined problem, and as such finding an optimal solution remains an open problem. Our gamut mapped results present differences with respect to the target BRDFs we are trying to represent. This is to be expected, since the target BRDFs lie outside the gamut, and therefore compromises need to be made when bringing them inside. These differences may be due to the inability of the inks to represent certain material properties (e.g., since there are no purely diffuse inks in our gamut, purely diffuse materials cannot be accurately represented), or to the optimization, since we cannot guarantee to find an optimal solution. Nevertheless, our approach yields better results in general than other state-of-the-art techniques.

Our attribute-based framework allows for versatility to achieve a variety goals, since different appearance properties can be preserved during the mapping process. As a consequence of this versatility, the particular choice of attributes may also have an influence on the final result, differing from the target BRDF. For example, *silvermetallicpaint* (Figure 5.10, top) benefits from the preservation of the *rough* attribute, since it is one of the main characteristics of the target BRDF, while preserving the *metallic-like* attribute instead would yield sharper reflections.

Finally, materials that lie far away from the gamut defined by the inks remain a challenging problem; in such cases our method may fail to faithfully reproduce the desired appearance, causing the resulting in-gamut BRDF to present visual differences with respect to the target BRDF. This behavior is similar in Pereira's work, suggesting that the limited gamut provided by the inks is the main cause for such differences in these cases. An example of this is depicted in Figure 5.12, showing also how the single-step state-of-the-art method fails. However, our result preserves the specular behavior better, thanks to our initial step in which we preserve the *metallic-like* attribute.

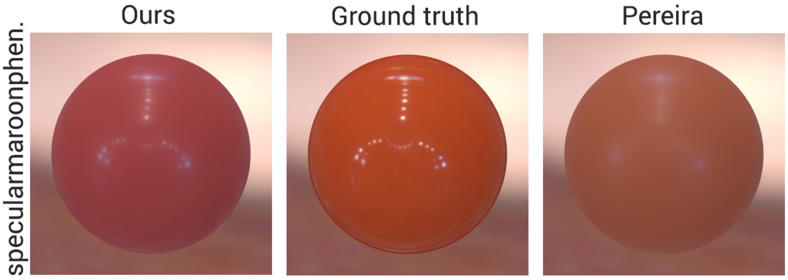


Figure 5.12: Limitations of current methods. If the BRDF lies very far away from the gamut (such as *specularmaroonphenolic* shown here) both our method and the single-step state of the art are unable to find a satisfying match in appearance. Here, our method does a reasonable job at preserving the specular behavior, but fails to accurately reproduce the diffuse color.

While currently users can choose to preserve different attributes with different weights, an interesting future line of work would be to conduct perceptual studies to analyze the influence of each attribute in the perceived appearance, in order to automatically assign weights to each attribute during the optimization process. Further investigation could also be devoted to optimizing for just a region of the image that contains most of the appearance information, as opposed to the whole image.

Part IV

Virtual reality

Virtual reality is rapidly entering the consumer market, however little is known about user behavior in this new environment. In the first chapter of this part we collect a large dataset of gaze samples and analyze user behavior in static omnistereo panoramas. In the second chapter, we gather gaze data from viewers watching VR videos containing different edits with varying parameters, and provide the first systematic analysis of viewers’ behavior and the perception of continuity in VR. In the third chapter, we present a method for adding parallax to virtual reality 360^{o} video visualization.

6. Saliency in VR: How do people explore virtual environments?

ABOUT THIS CHAPTER

The work presented in this chapter was presented at the *IEEE VR* 2018 conference, and published in *IEEE Transactions on Visualization and Computer Graphics*. In this work we investigate how people explore immersive virtual environments. To further our understanding of viewing behavior and saliency in virtual reality, we capture and analyze gaze and head orientation data of users exploring static omnistereo panoramas. This work was developed in collaboration with *Stanford University*, and I shared the first authorship with Vincent Sitzmann. My contribution to this work lies mainly in the first part, where we provide a thorough analysis of our data, and in the proof-of-concept of some of the applications derived from these insights.

V. Sitzmann*, A. Serrano*, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein.
*These authors contributed equally to the work.
Saliency in VR: How do people explore virtual environments?
IEEE Transactions on Visualization and Computer Graphics, Vol. 24 (4), 2018.

6.1 Introduction

Virtual reality (VR) systems provide a new medium that has the potential to have a significant impact on our society. The experiences offered by these emerging systems are inherently different from radio, television, or theater, opening new directions in research areas such as cinematic VR capture [12], interaction [309], or content generation and editing [234, 292]. However, the behavior of users who visually explore immersive VR environments is not well understood, nor do statistical models exist to predict this behavior. Yet, with unprecedented capabilities for creating synthetic immersive environments, many important questions arise. How do we design 3D scenes or place cuts in VR videos? How do we drive user attention in virtual environments? Can we predict visual exploration patterns? How can we efficiently compress cinematic VR content?

To address these and other questions from first principles, it is crucial to understand how users explore virtual environments. In this work, we take steps towards this goal. In particular, we are interested in quantifying aspects of user behavior that may be helpful in predicting exploratory user behavior in static and dynamic virtual environments observed from a fixed viewpoint. A detailed understanding of visual attention in VR would not only help answer the above questions, but also inform future designs of user interfaces, eye tracking technology, and other key aspects of VR systems.

A crucial requirement for developing an understanding of viewing behavior in VR is access to behavioral data. To this end, we have performed an extensive study, recording 1980 head and gaze trajectories from 169 people in 22 static virtual environments, which are represented as stereoscopic omni-directional panoramas. Data is recorded using a head-mounted display (HMD) in both standing and seated conditions (*VR* condition and *VR seated* condition), as well as for users observing the same scenes in mono on a desktop monitor for comparison (*desktop* condition).

We analyze the recorded data and discuss important insights, such as the existence of a fixation bias, the mean time until a static stereo panorama can be considered to be fully explored by users, or the existence of two apparent modes in viewer behavior, attention and re-orientation (see Sec. 6.4 for more details). We then leverage our data to evaluate *existing* saliency predictors, designed for narrow field of view video, in the context of immersive VR, and show how these

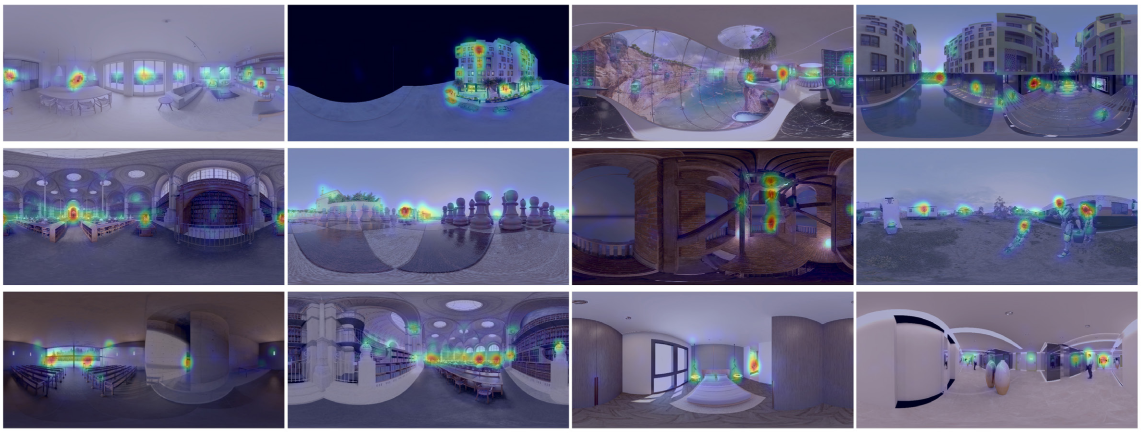


Figure 6.1: A representative subset of the 22 panoramas used to analyze how people explore virtual environments from a fixed viewpoint. We recorded almost two thousand scanpaths of users exploring these scenes in different immersive and non-immersive viewing conditions. We then analyzed this data, and provide meaningful insights about viewers' behavior. We apply these insights to VR applications, such as saliency prediction (shown in the image as overlaid heatmaps), VR movie editing, panorama thumbnail generation, panorama video synopsis, and saliency-aware compression of VR content.

can be adapted to VR applications. Saliency prediction is a well-explored topic and many existing models are evaluated by the MIT Saliency Benchmark [46]. However, these models assume that users sit in front of a screen while observing the images – ground truth data is collected by eye trackers recording precisely this behavior. VR is different from traditional 2D viewing in that users naturally use both significant head movement and gaze to visually explore scenes. We show that this leads to a fixation bias around the equator that is not observed in conventional viewing conditions. Figure 6.1 shows panoramic views of some of our 22 scenes with superimposed saliency computed from the recorded scan paths in the VR condition. Apart from saliency, we offer several other example applications that are directly derived from our findings. Specifically, our contributions are:

- We record and provide an extensive dataset of visual exploration behavior in stereoscopic, static omni-directional stereo (ODS) panoramas. The dataset contains head orientation and gaze direction, and it captures several different viewing conditions. Scenes, data, and code for analysis (Sec. 6.3) are available online in our project website[15]
- We provide low-level and high-level analysis of the recorded dataset. We derive relevant insights that can be crucial for predicting saliency in VR and other VR applications, such as the existence of an attention bias in VR scenes or differences in head and gaze movement statistics when fixating (Sec. 6.4)
- We evaluate existing saliency predictors with respect to their performance in VR applications. We show how to tailor these predictors to ODS panoramas and demonstrate that saliency prediction from head movement alone performs on par with state-of-the-art saliency predictors for our scenes (Sec. 6.5)
- We demonstrate several applications of this saliency prediction, including automatic panorama thumbnails, VR video synopsis, compression, and VR video cuts (Sec. 4.7)

[15] https://vsitzmann.github.io/vr-saliency

6.2 Related work

Modeling human gaze behavior and predicting visual attention has been an active area of vision research. In their seminal work, Koch and Ullman [167] introduced a model for predicting salient regions from a set of image features. Motivated by this work, many models of visual attention have been proposed throughout the last three decades. Most of these models are based on bottom-up, top-down, or hybrid approaches. Bottom-up approaches build on a combination of low-level image features, including color, contrast, or orientation [59, 150, 165, 197] (see Zhao and Koch [356] for a review). Top-down models take higher-level knowledge of the scene into account such as context or specific tasks [114, 154, 158, 196, 321]. Recently, advances in machine learning and particularly convolutional neuronal networks (CNNs) have fostered the convergence of top-down and bottom-up features for saliency prediction, producing more accurate models [145, 191, 244, 332, 357]. Jiang et al. [155] proposed a new methodology to collect attentional data on scales sufficient for these deep learning methods. Volokitin et al. [329] used features learned by CNNs to predict when saliency maps predicted by a model will be accurate and when fixations will be consistent among human observers. Significant prior work explored rigorous benchmarking of saliency models, the impact of the metric on the evaluation result, and shortcomings of state-of-the-art models at the time [33, 45, 268]. Recent work also attempts to extend CNN approaches beyond classical 2D images by computing saliency in more complex scenarios such as stereo images [68, 122] or video [51, 187]. A related line of research is devoted to modeling the gaze scanpath followed by subjects, i.e., the temporal evolution of the viewer's gaze [156, 183]. Marmitt et al. [213] developed a metric to evaluate predicted scanpaths in VR and showed that predictors built for classic viewing conditions perform significantly worse in VR. Building on the rich literature in this area, we explore user behavior and visual attention in immersive virtual environments, which can help build similar models for VR.

What makes VR different from desktop viewing conditions is the fact that head orientation is used as a natural interface to control perspective (and in some cases navigation as well [322]). The interactions of head and eye movements are complex and neurally coupled, for example via the vestibulo-ocular reflex [179]. Koehler et al. [168] showed that saliency maps can differ depending on the instructions given to the viewer. For more information on user behavior in VR, we refer to Ruhland et al. [271], who provide a review of eye gaze behavior, and Freedman [105], who discusses the mechanisms that characterize the coordination between eyes and head during visual orienting movements. With the data recorded in this project, we observe the vestibulo-ocular reflex and other interesting effects. In the main text and Appendix C, we provide an extensive analysis of the user data, and derive statistics describing many low-level aspects of viewing behavior. We hope that this analysis will be useful for basic vision research.

Recent work of Nakashima et al. [227] is closely related to some aspects of our work. They propose a head direction prior to improve accuracy in saliency-based gaze prediction through simple multiplication of the gaze saliency map by a Gaussian head direction bias. Concurrent work by Upenik et al. [325] explores visual attention in VR solely by tracking head orientation. The data collected in this work and in-depth analyses augment prior work in this field, and may allow for future data-driven models for visual behavior to be learned.

Finally, gaze tracking has found many applications in VR user interfaces [315] and gaze-contingent displays [90, 243, 303]. The ability to predict viewing behavior would be helpful for all of these applications. For example, gaze-contingent techniques may become possible without dedicated gaze trackers, which are currently expensive and not widely available. Moreover, techniques for editing VR content are starting to emerge [234, 292]. The understanding of user behavior we aim to develop in this work could also influence these and other tools for content creation.

6.3 Recording head orientation and gaze

In this section, we summarize our efforts towards recording a dataset that contains head orientation and gaze direction for users watching stereoscopic VR panoramas in several different viewing conditions; we provide additional details in the Appendix C. These data form the basis of a statistical analysis of viewing behavior (Sec. 6.4), as ground truth for saliency prediction (Sec. 6.5), and also as reference saliency for several higher-level applications (Sec. 4.7).

6.3.1 *Data capture*

STIMULI For the experiments reported in this work, we used 22 high-resolution omni-directional stereo panoramas (see Figure 6.1 and Appendix C). We opt for a fixed viewpoint because for the subsequent analyses it is crucial that subjects see the exact same content. Further, in a 3D scenario the variability is likely to be much higher, requiring extremely large numbers of subjects to draw significant conclusions. The scenes include (14) indoor and (8) outdoor scenarios and do not contain landmarks that may be recognized by the users. For each scene we explore different conditions, which limits the number of scenes we can have with the experiment size remaining tractable. With the current stimuli and conditions, we have collected nearly 2,000 trajectories from 169 viewers. All scenes are computer generated by artists; we received permission to use them for this study.

CONDITIONS We recorded users observing the 22 panoramas in three different conditions: in a standing position using a head-mounted display (i.e., the *VR* condition), seated in a non-swivel chair using a head-mounted display (i.e., the *VR seated* condition, making it more difficult to turn around), and seated in front of a desktop monitor (i.e., the *desktop* condition). In the *desktop* condition, the scenes are monoscopic, and users navigate with a mouse. For each scene, we tested four different starting points, spaced at 90° longitude, which results in a total of 264 conditions. These starting points were chosen to cover the entire longitudinal range, while keeping the number of different conditions tractable. We chose not to randomize the starting point over the whole latitude (and rather select randomly from four fixed ones) to limit the number of conditions while being able to analyze the influence of the starting point (Sec. 6.4.5 and Appendix C). Complete randomization over the starting point could be of interest for future studies.

PARTICIPANTS For the *VR* condition, we recorded 122 users (92 male, 30 female, age 17-59). The experiments with the *VR seated* condition were performed by 47 users (38 male, 9 female, age 17-39). Users were asked to first perform a stereo vision (Randot) test to quantify their stereo acuity. For *desktop* experiments, we recruited 44 additional participants (27 male, 17 female, age 18-33). All participants reported normal or corrected-to-normal vision.

PROCEDURE All VR scenes were displayed using an Oculus DK2 head-mounted display, equipped with a pupil-labs[16] stereoscopic eye tracker recording at 120 Hz. The DK2 offers a field of view of $95 \times 106°$. The Unity game engine was used to display all scenes and record head orientation while the eye tracker collected gaze data on a separate computer. Users were instructed to freely explore the scene and were provided with a pair of earmuffs to avoid auditory interference. Scenes and starting points were randomized, while ensuring that each user would only see the same scene once from a single random starting point. Each user was shown 8 scenes. Each scene in a certain condition was shown to the user during 30 seconds, while the total time per user that the experiment took, including calibration and explanation, was approximately 10 minutes.

We modeled the *desktop* condition after typical, mouse-controlled desktop panorama viewers on the web (i.e., YouTubeVR or Facebook360). Users sat 0.45 meters away from a 17.3″ monitor with a resolution of 1920×1080 px, covering a field of view of $23 \times 13°$. We used a Tobii EyeX eye tracker

[16] https://pupil-labs.com

with an accuracy of 0.6° at a sampling frequency of 55 Hz [110]. The image viewer displayed a rectilinear projection of a $97 \times 65°$ viewport of the panorama. To keep the field of view consistent, no zooming was possible. We instructed the users on how to use the image viewer, before showing the 22 scenes for 30 seconds each. In this condition, we only collected gaze data since users rarely re-orient their head. Instead, we recorded where the users interactively place the virtual camera in the panorama as a proxy for head orientation.

6.3.2 *Data processing*

To identify fixations, we transformed the normalized gaze tracker coordinates to latitude and longitude in the 360° panorama. This is necessary to detect users fixating on panorama features while turning their head. We used thresholding based on dispersion and duration of the fixations [272]. For the VR experiments, we set the minimum duration to 150 ms [272] and the maximum dispersion to 1° [30]. For the desktop condition, we found the Tobii EyeX eyetracker to be more noisy than the PupilLabs eyetracker. Thus, we first smoothed this data with a running average of 2 samples, and detected fixations with a dispersion of 2°. We counted the number of fixations at each pixel location in the panorama. Similar to Judd et al. [158], we only consider measurements from the moment where user's gaze left the initial starting point to avoid adding trivial information. We convolved these fixation maps with a Gaussian with a standard deviation of 1° of visual angle, the area of the field of view seen sharply on the fovea of the user, to yield continuous saliency maps [181].

6.4 Understanding viewing behavior in VR

With the recorded data, we can gather insights and investigate a number of questions about the behavior of users exploring virtual environments. In the following, we analyze both low-level characteristics, such as duration of the fixations and speed of gaze, and higher-level characteristics, such as the influence of the content or characteristics of the scene.

6.4.1 *Is viewing behavior similar between users?*

We first want to assess whether viewing behavior between users is similar; this is also indicative of how robust our data is, and thus how much we can rely on it to draw conclusions. To answer this, we analyze the agreement between users. Specifically, we compute the *inter-observer congruency* metric by means of a *receiver operating characteristic* curve (ROC) [181, 321]. This metric calculates the ability of the i^{th} user to predict a *ground truth saliency map*, which is computed from the fixations of all the other users averaged. A single point in the ROC curve is computed by finding the top n% most salient regions of the ground truth saliency map (leaving out the i^{th} user), and then calculating the percentage of fixations of the i^{th} user that fall into these regions. We show the average ROC for all the 22 scenes in Figure 6.2 (left), compared with chance (the individual ROCs for each scene are depicted in light gray). The fast convergence of these curves to the maximum rate of 1 indicates a strong agreement, and thus similar behavior, between users for each of the scenes tested. 70% of all fixations fall within the 20% most salient regions. These values are comparable to previous studies viewing regular images on a display [181].

6.4.2 *How different is viewing behavior for the three conditions?*

An important question to ask is whether viewing behavior changes when exploring a scene under different conditions. Visual inspection of our three conditions (*VR*, *VR seated*, and *desktop*) shows a high similarity between the saliency maps (see Appendix C). For a quantitative evaluation of the similarity of saliency maps (here, and in the rest of the chapter), we use the Pearson correlation

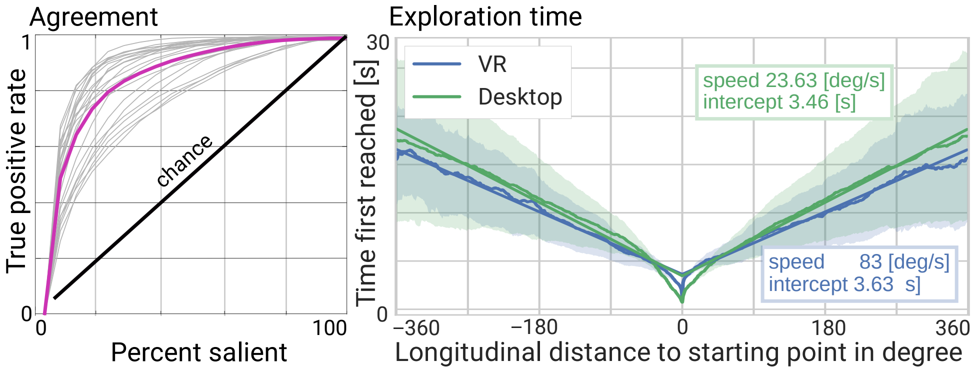


Figure 6.2: *Left:* ROC curve of human performance averaged across users (magenta) and individual ROCs for each scene (light gray). The fast convergence to the maximum is indicative of a strong agreement between users. *Right:* Exploration time computed as the average time until a specific longitudinal offset from the starting point is reached.

(CC) score, which is a widely used metric in saliency map prediction [45]. It ranges from -1 (perfectly inversely correlated) to 1 (perfectly correlated). The high similarity is confirmed by a median CC score of 0.80 when comparing the *VR* and the *VR seated* conditions, and 0.76 when comparing the *VR* and the *desktop* conditions. The latter is a significant insight: since desktop experiments are much easier to control, it may be possible to use these for collecting adequate training sets for data-driven saliency prediction in future VR systems. Given this similarity, we report only the results of the *VR* (standing) condition throughout the remainder of this chapter, unless a significant difference is found, and refer the reader to Appendix C for the *VR seated* and *desktop* conditions.

6.4.3 *Is there a fixation bias in VR?*

Several researchers have reported a strong bias for human fixations to be near the center, when viewing regular images [158, 238]. A natural question to ask is whether a similar bias exists in VR. Similar to Judd et al. [158], we calculate the average of all 22 saliency maps, and filter out fixations within the close vicinity (20° longitude) of the starting point. The resulting data indicates that users tend to fixate around the equator of the panoramas, with very few fixations in latitudes far from it. To quantify this *equator bias*, we marginalize out the longitudinal component of the saliency map, and fit a Laplace distribution—with location parameter μ and diversity β—to the latitudinal component (this particular distribution yielded the best match among several tested distributions). Figure 6.3 depicts the average saliency map, as well as our Laplacian fit to the latitudinal distribution and its parameters, for both the *VR* and the *desktop* conditions. While the mean is almost identical, the equator bias for the desktop condition has a lower diversity. As discussed in Section 6.5, this Laplacian equator bias is crucial for predicting saliency in VR.

Note that most of the scenes in our study have a clear horizon line, which may have influenced the observed equator bias along with viewing preferences, kinematic constraints, as well as the static nature of the scenes. However, most virtual environments share this type of scene layout, so we believe our findings generalize to a significant fraction of this type of content. Further, even for scenes with content scattered along different latitudes (see, e.g., *scene 16* in Appendix C, displaying very few salient areas near the poles), we observed an equator bias. Nevertheless, different tasks or scenarios, such as gaming, may influence this bias.

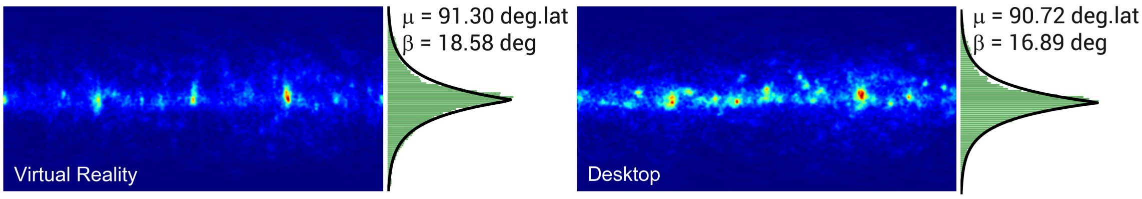


Figure 6.3: Average saliency maps computed with all the scenes for both the *VR* (left) and the *desktop* (right) conditions. These average maps demonstrate an "equator bias" that is well-described by a Laplacian fit modeling the probability of a user fixating on an object at a specific latitude.

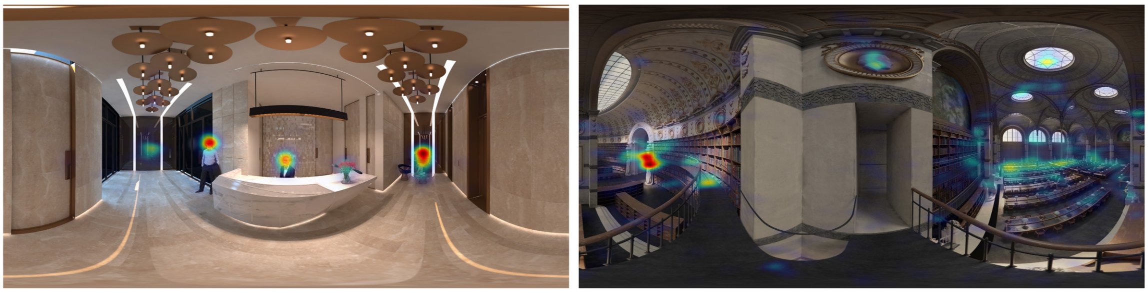


Figure 6.4: Saliency maps presenting the lowest (*left*) and highest (*right*) entropy in our dataset. Saliency maps with low entropy have very defined salient regions while in maps with high entropy fixations are scattered all over the scene.

6.4.4 *Does scene content affect viewing behavior?*

A fundamental issue when analyzing viewing behavior is the potential influence of scene content. This is of particular relevance for content creators; since in a VR setup the viewer has control over the camera, this analysis can help address the key challenge of predicting user attention.

To characterize scene content in a manner that enables insightful analysis, we rely on the distribution of salient regions in the scene, in particular on the *entropy* of the saliency maps. A high entropy results from a large number of similarly salient objects distributed throughout the scene, causing users' fixations to be scattered all over the scene; a low entropy results from a few salient objects that capture all the viewer's attention. Figure 6.4 shows the saliency maps of the scenes with lowest and highest entropy in our dataset.

Our entropy is computed as the Shannon entropy of the *ground truth saliency map*, computed, per scene, from the average of all users [158]. The entropy is given by: $-\sum_{i=1}^{N} s_i^2 log(s_i^2)$, with s being the ground truth saliency, and N the number of pixels. We consider two entropy levels, low and high, which we term $\{E_0, E_1\}$, respectively. Since a clear threshold for classifying each scene according to its entropy does not exist, we take a conservative approach and analyze only the four scenes with highest and the four with lowest entropy, for a total of eight scenes.

6.4.4.1 *Viewing behavior metrics*

Measuring viewing behavior in an objective manner is not a simple task. First, we define *salient regions* as the 5% most salient pixels of a scene. Figure 6.5 shows a saliency map and the resulting salient regions computed with this criterion. We then rely on three metrics recently proposed by Serrano et al. [292] in the context of gaze analysis for VR movie editing (time to reach a salient region (*timeToSR*), percentage of fixations inside the salient regions (*percFixInside*), and number of fixations (*nFix*), which are summarized in Appendix C), and propose a fourth, novel one, tailored (but not limited) to quantifying the degree of exploration over time in static 360° panoramas:

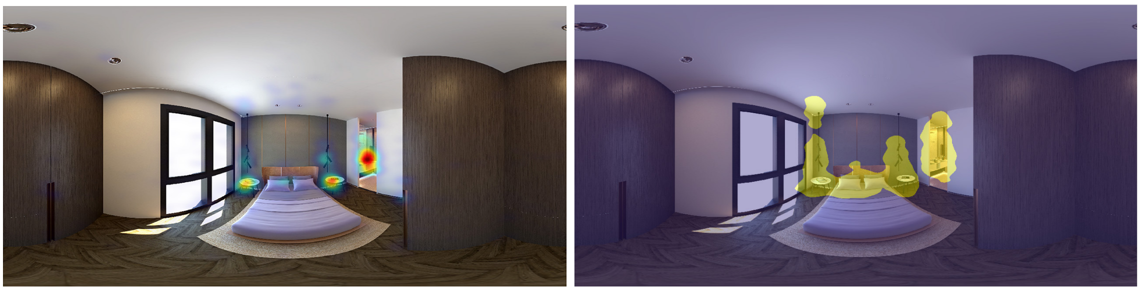


Figure 6.5: Salient region computation. *Left:* Ground truth saliency map for a sample scene. *Right:* Corresponding salient regions (yellow) computed by thresholding the 5% most salient pixels of the scene.

CONVERGENCE TIME (*convergtime*) For every scene, we obtain the per-user saliency maps at different time steps, and compute the similarity (CC score) with the fully-converged saliency map. We plot the temporal evolution of this CC score, and compute the area under this curve. This metric represents the temporal convergence of saliency maps; it is inversely proportional to how long it takes for the fixation map during exploration to converge to the ground truth saliency map.

6.4.4.2 *Analysis*

We first test for independence of observations performing a Wald's test (please refer to Appendix C). Based on its results, we employ ANOVA when analyzing *percFixInside*, since the samples are considered to be independent, and report significance values obtained from multilevel modeling for the other three metrics.

We find that the *entropy* of the scene has a significant effect on *nFix* ($p < 0.001$), *timeToSR* ($p < 0.001$), *percFixInside* ($p = 0.022$), and *convergTime* ($p < 0.001$). Specifically, on scenes with low entropy (E_0), the time to reach a salient region (*timeToSR*) is lower. This may be counter-intuitive, since high entropy scenes contain a larger number of salient regions and thus it would be easier to reach one. Interestingly, though, our results indicate that the viewer explores the scene faster in cases of low entropy, quickly discarding non-salient regions, and that their attention gets directed towards the few salient regions faster. This hypothesis is further supported by the behavior of the *convergTime* metric, which shows that scenes with low entropy do converge faster, and is consistent with the number of fixations, and fixations inside the salient region (*nFix* and *percFixInside*): both are higher for low entropy scenes, indicating that users pay more attention to salient regions when such regions are less, and more concentrated.

6.4.5 *Does the starting point affect viewing behavior?*

We also evaluate whether the starting viewport conditions the final saliency map for a given scene: For each scene, we compute the similarity between the final saliency map of the i^{th} viewport and the other three, using again the CC score. We obtain a median CC score of 0.79, which indicates that the final saliency maps after 30 seconds, starting from different viewports, converge and are very similar. Additional analysis on the influence of the viewport, including also a state sequence analysis [106, 292], can be found in Appendix C.

6.4.6 *How are head and gaze statistics related?*

Many additional insights can be learned from our data, which may be useful for further vision and cognition research, or in applications that require predicting gaze or saliency in VR (see also Section 6.5). First, we evaluate the speed with which users explore a given scene. Figure 6.2 (right)

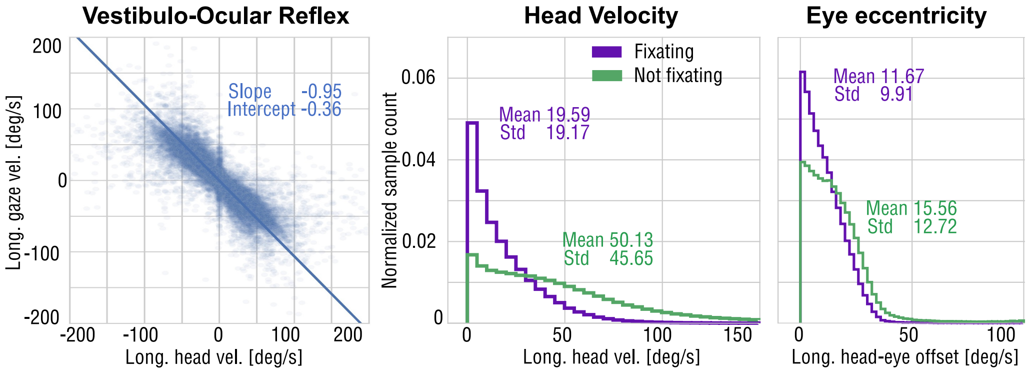


Figure 6.6: *Left:* the vestibulo-ocular reflex demonstrated by an inverse linear relationship of gaze and head velocities. *Middle and right:* distributions of longitudinal head velocity and longitudinal eye eccentricity, respectively, while fixating and while not fixating.

shows this *exploration time*, which is the average time that users took to move their eyes to a certain longitude relative to their starting point. On average, users fully explored each scene after about 19 seconds. Indeed, after this time, all saliency maps in our dataset have converged to a CC score of at least 0.8 as compared to their final state. These results suggests that an experimental time of 20 seconds is sufficient to capture fixations in static stereo panoramas.

In our experiments, the mean number of fixations across scenes is 55.35 ± 12.85 for the VR condition and 49.68 ± 15.04 for the desktop condition. The mean duration of fixations in the *VR* condition is 266 ms ± 132, and 253 ms ± 124 in the *desktop* condition. This is in the range reported for traditional screen viewing conditions [272]. The mean gaze direction relative to the head orientation across scenes is 13.85° ± 11.73, which is consistent with the analysis performed by Kollenberg et al. [170].

We have also identified the *vestibulo-ocular reflex* [179] in our data. This reflexive mechanism moves the eyes contrary to the head movement, in order to stabilize the line of sight and thus improve vision quality. Figure 6.6 (left) shows the expected inverse linear relationship between head velocity and relative gaze velocity when fixating. Given this observation, we further analyze the interaction between eye and head movements when shifting to a new target. We offset in time head and gaze acceleration measurements relative to each other, and compute the cross-correlation for different temporal shifts. Our data reveals that head follows gaze with an average delay of 58 ms, where the largest cross-correlation is observed, which is consistent with previous work [88, 105].

It is well-known that gaze velocities differ when users fixate and when they do not [272]. We look at whether this is also the case for head velocities, since they could then act as a rough proxy for fixation classification. Figure 6.6 (middle) shows that users move their head at longitudinal velocities significantly below the average head speed when they are fixating, and above average when they are not. Further, Figure 6.6 (right) shows that the longitudinal rotation angle of the eyes relative to the head orientation (eye eccentricity) is significantly smaller when users are fixating. According to this data, users appear to behave in two different modes: *attention* and *re-orientation*. Eye fixations happen in the attention mode, when users have "locked in" on a salient part of the scene, while movements to new salient regions happen in the re-orientation mode. Being able to identify such modes in real time, from either head or gaze movement, can be very useful for interactive applications. Further results for the different conditions, and for the latitudinal direction, can be found in Appendix C. Finally, this data and findings can be leveraged for *time-dependent* and *head-based* saliency prediction, as we will show in Sections 6.5.2 and 6.5.3.

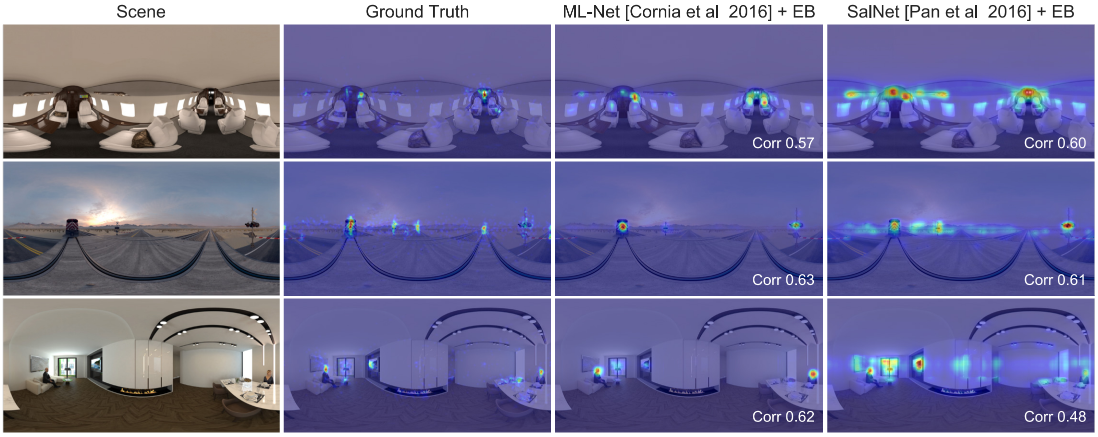


Figure 6.7: Saliency prediction for omni-directional stereo panoramas. Existing saliency predictors can be applied to spherical panoramas after they are projected onto a plane, here performed with the patch-based method described in the text. These methods tend to over-predict saliency near the poles. By multiplying the predicted saliency map by the longitudinal equator bias (EB) derived in the previous section, we achieve a good match between ground truth (center left) and predicted saliency (right). Note that this procedure could be applied to any saliency predictor; we chose two top-scoring predictors as an example.

6.5 Predicting saliency in VR

In this section, we show how existing saliency prediction models can be adapted to VR using insights of our data analysis, such as the equator bias. Then, we ask whether the problem of time-dependent saliency prediction is a well-defined one that can be answered with sufficient confidence. Finally, we analyze how well head movement alone, for example captured with inertial sensors, can predict saliency without knowing the exact gaze direction.

6.5.1 *Predicting saliency maps*

Instead of learning VR saliency models from scratch, we ask whether existing models could be adopted to immersive applications. This would be ideal, because many saliency predictors for desktop viewing conditions already exist, and advances in that domain could be directly transferred to VR conditions. The fact that gaze statistics are closely related in VR and in traditional viewing (Section 6.4.6) is indicative of the fact that existing saliency models may be adequate, at least to some extend, to VR. In this context, two primary challenges arise: (i) mapping a 360° panorama to a 2D image (the required input for existing models) distorts the content due to the projective mapping from sphere to plane; and (ii) head-gaze interaction may require special attention for saliency prediction in VR. We address both of these issues in the following.

Which projection is best?

Before running a conventional saliency predictor on a spherical panorama or parts of it, the image has to be projected into a plane. Different projections would naturally result in different types of distortions that may affect the saliency predictor. For an equirectangular projection, for example, we expect large distortions near the poles. A cube map projection may result in discontinuities between some of the cube's faces. Alternatively, smaller patches can be extracted from the panorama, saliency prediction applied to each of them projected onto a plane, and the result stitched together and blended into a saliency panorama. The latter, patch-based approach would result in the least amount of geometrical distortions, but it is also the most computationally expensive approach and it gives up global context for the saliency prediction.

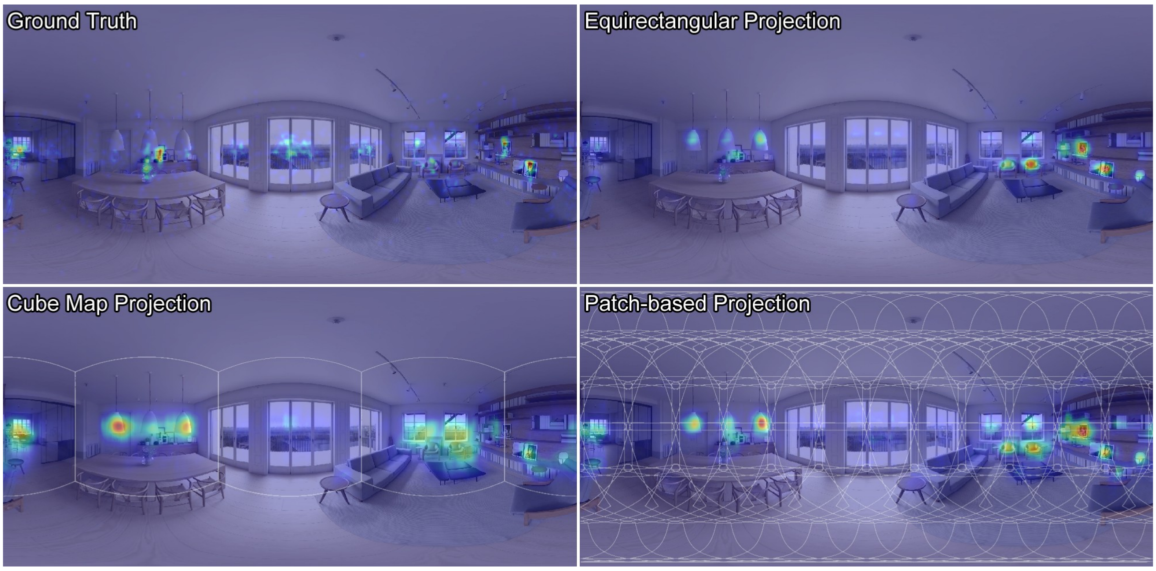


Figure 6.8: Comparison of saliency prediction using different projections from sphere to plane. After applying the equator bias, all three projection methods result in comparable saliency maps for this example.

	Equirectangular	Cube Map	Patch Based
Without Equator Bias	μ =0.48	μ = 0.37	μ =0.43
With Equator Bias	μ =**0.50**	μ =0.44	μ =0.49

Table 6.1: Quantitative evaluation of three different projection methods with and without equator bias. We list the mean CC score for all 22 VR scenes used in this study. Applying the equator bias significantly improves the quality of all approaches. Distortions of the equirectangular projection near the poles do not affect saliency prediction as much as the shortcomings of other types of projection after the equator bias is applied.

In Figure 6.8 and Table 6.1 we compare saliency prediction using all three projection methods qualitatively and quantitatively. For each projection, we compute a saliency map using the state-of-the-art ML-Net saliency predictor [70], and then optionally multiply it by the latitudinal equator bias we derived in Section 6.4.3. We incorporate the equator bias in a multiplicative manner. This only increases the weight of areas that the saliency predictor has found to be potentially salient, while an additive bias would increase saliency of all points around the equator. Alternatively, it could also be incorporated by addition and re-normalization. Figure 6.8 shows an example saliency map predicted on the three different sphere projections after applying the equator bias. We also compare the average CC score for all three projection methods and all 22 scenes in Table 6.1. Quantitatively, saliency computed directly on the equirectangular projection with the equator bias applied not only performs best but it is also the fastest of the three approaches. The benefit of applying the equator bias is smaller for the equirectangular projection than for the other two projections. This may be because the distortions at the poles introduced by the projection may naturally lead to less saliency predicted at the poles than in the cube map and patch-based approaches. While this seems to make this prediction method competitive even without the equator bias, it may lead to inferior generalization as compared to an explicit modeling. Since the equirectangular and patch-based methods using the equator bias perform almost on par, in the following, we use the patch-based method when processing time is not critical, since it is not susceptible to projective distortions.

	EB	ML-Net + EB	SalNet + EB
VR	$\mu = 0.34 \pm 0.13$	$\mu = \mathbf{0.49} \pm 0.11$	$\mu = 0.47 \pm 0.13$
Desktop	$\mu = 0.37 \pm 0.11$	$\mu = \mathbf{0.57} \pm 0.11$	$\mu = 0.52 \pm 0.12$

Table 6.2: Quantitative comparison of predicted saliency maps using a simple equator bias (EB), and two state-of-the-art models together with the EB. Numbers show average mean and standard deviation of CC scores, for each scene, between prediction and ground truth recorded from users exploring 22 scenes in the VR and desktop conditions. The proposed patch-based method was used to predict the saliency maps for both predictors.

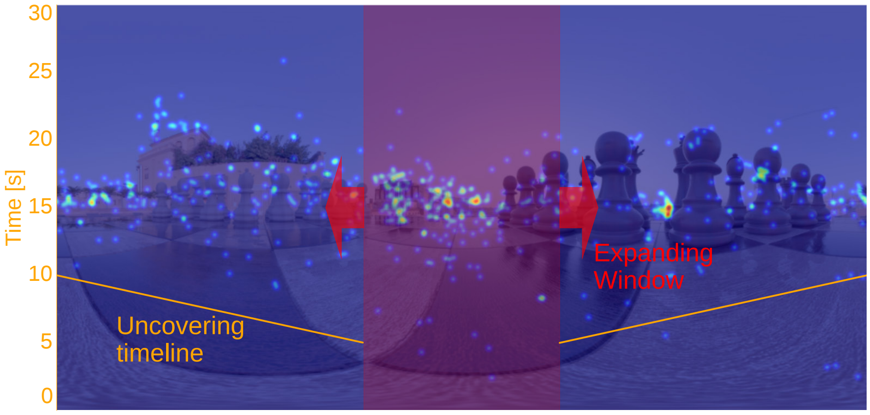


Figure 6.9: Time-dependent saliency prediction by uncovering the converged saliency map with the average exploration speed determined in Section 6.4.

Which predictor is best?

The fact that existing saliency predictors seem to apply to VR scenarios is important, because rapid progress is being made for saliency prediction with images and videos. Advances in those domains could directly improve saliency prediction in VR. Here, we further evaluate several different existing predictors both quantitatively and qualitatively.

Table 6.2 lists mean and standard deviation of the CC score for all 22 scenes in the VR condition, and for users exploring the same scenes in the desktop condition. These numbers allow us to analyze how good and how consistent across scenes a particular predictor is. We test the equator bias by itself as a baseline, as well as two of the highest-ranked models in the MIT benchmark where source code is available: ML-Net [70] and SalNet [244], together with the equator bias. We see that the two advanced models perform very similar and do much better than the equator bias alone. We also see that both of these models predict viewing behavior in the desktop condition better than for the VR condition. This makes sense, because the desktop condition is what these models were trained for originally. In Figure 6.7 we also compare qualitatively the saliency maps of three scenes recorded under the VR condition (all scenes in Appendix C).

6.5.2 *Can time-dependent saliency be predicted with sufficient confidence?*

Virtual environments impose viewing conditions much different from those of conventional saliency prediction. Specifically, the question of temporal evolution arises: for users starting to explore the scene at a given starting point, is it possible to predict the probability that they fixate at specific coordinates at a time instant t? This problem is also closely related to scanpath prediction. We use data from Section 6.4 to build a simple baseline model for this problem: Figure 6.2 (right) shows an estimate for when users reach a certain longitude on average. We can thus model the time-dependent saliency map of a scene with an initially small window that grows larger over time to progressively uncover more of a converged (predicted or ground truth) saliency map. The part of the saliency map within this window is the currently active part, while the parts outside this window are set to zero. The left and right boundaries of the window are widened with the speed predicted in Figure 6.2 (right).

Figure 6.9 visualizes this approach. We generate the time-dependent saliency maps from the converged ground truth maps for all 22 scenes and compare them with the actual ground truth at each timestep. We use the fully-converged saliency map as a baseline. The time-dependent, constructed saliency maps model the recorded data better than the converged saliency map within the first 6 seconds. Subsequently, they perform slightly worse until the converged map is fully uncovered after about 10 seconds, and the model is thus identical to the baseline. Our simple time-dependent model achieves an average CC score of 0.57 over all scenes, viewports, and the first 10 seconds (uncovering the ground truth saliency map), while using the converged saliency map as a predictor yields a CC of just 0.47.

Although this is useful as a first-order approximation for time-dependent saliency, there is still work ahead to adequately model time-dependent saliency over prolonged periods. In fact, due to the high inter-user variance of recorded scanpaths[17], the problem of predicting time-dependent saliency maps may not be a well-defined one. Perhaps a real-time approach that would use head orientation measured by an inertial measurement unit (IMU) to predict where a specific user will look next could be more useful than trying to predict time-dependent saliency without any knowledge of a specific user.

6.5.3 *Can head orientation be used for saliency prediction?*

The analysis in Section 6.4 indicates a strong correlation between head movement and gaze behavior in VR. In particular, Figure 6.6 (middle) shows that fixations usually occur with low head velocities (except for the vestibulo-ocular reflex). This insight suggests that an approximation of a saliency map may be obtained from the longitudinal head velocity alone, e.g. measured by an IMU, without the need for gaze tracking.

We validate this hypothesis by counting the number of measurements at pixel locations where the head speed falls below a threshold of 19.6 $^\circ/s$ for all experiments in the VR condition. We then blur this information with a Gaussian kernel of size 11.7° of visual angle, to take into account the mean eye offset while fixating (Figure 6.6, right). Qualitative results are shown in Appendix C. For a quantitative analysis, we compute the CC score between these *head saliency maps* and the ground truth and compared it with the results obtained from the predictors examined in Table 6.2. Our CC score of 0.50 places our approximation on par with the performance of both saliency predictors tested; this is a positive and interesting result, given the fact that no gaze information is used at all. Head saliency maps could therefore become a valuable tool to analyze the approximate regions that users attend to from IMU data alone, without the need for additional eye-tracking hardware.

6.6 Applications

In this section, we outline several applications for VR saliency prediction. Rather than evaluating each of the applications in detail and comparing extensively to potentially related techniques, the goal of this section is to highlight the importance and utility of saliency prediction in VR for a range of applications with the purpose of stimulating future work in this domain.

6.6.1 *Automatic alignment of cuts in VR video*

How to place cuts in VR video is a question that was recently addressed by Serrano et al. [292].

In a number of situations, alignment of the objects of interest before and after the cut is a safe

[17] While converged saliency maps show a high inter-user agreement (Section 4.6.2), this is not necessarily the case for scanpaths, and thus for time-dependent saliency.

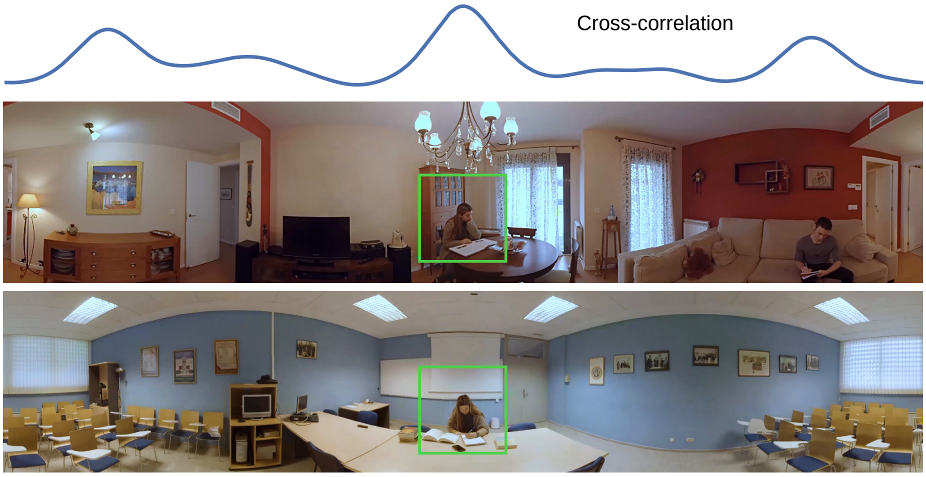


Figure 6.10: Automatic alignment of cuts in VR video. To align two video segments, we can maximize the correlation between the saliency maps of the last frame in the first segment and the first frame of the second one. The cross-correlation accounting for all horizontal shifts is shown on top of this example, which has been automatically aligned with the proposed algorithm.

assumption, since it facilitates the viewer to "lock in" on the action immediately after the cut. The proposed saliency prediction facilitates automatic alignment of such cuts. We show in Figure 6.10 that predicted saliency maps can be used to align VR video before and after a cut by shifting the cuts in the longitudinal direction such that the Pearson CC of the predicted saliency maps is maximized. We use the 72 scenes provided by Serrano et al. [292], which were manually aligned to overlapping regions of interest (ROI) before and after a cut. However, in some there are multiple ROIs, and thus multiple meaningful alignments possible. We predict saliency maps before and after the cut using the predictor described as performing best in Section 6.5.1 (i.e., ML-Net with equator bias on equirectangular projection), and then shift the saliency map after the cut with respect to the saliency map before the cut such as to maximize the Pearson correlation. For the scenes with one ROI visible before and after the cut, the median error of our method with respect to the manually aligned results is 2.11°, which mildly increases to 9.14° if we include the scenes with two ROIs in the same field of view. Qualitative analysis shows that the alignments are meaningful and succeed to align salient regions, however, performance is strongly dependent on the quality of the saliency predictor used. This indicates that saliency-based automatic alignment of video cuts is a useful way to guide users when editing VR videos, suggesting good initial alternatives, but it may not be able to completely replace user interaction. Full alignment results can be found in Appendix C.

6.6.2 *Panorama thumbnails*

Extracting a small viewport that is representative of a panorama may be helpful as a preview or thumbnail. However, VR panoramas cover the full sphere and most of the content may not be salient at all. To extract a thumbnail that remains representative of a scene in more commonly used image formats and at lower resolutions, we propose to extract the gnomonic, or rectilinear patch of the panorama that maximizes saliency within. To this end, we predict the saliency map of the entire panorama as discussed in Section 6.5.1. Then, we use an exhaustive search for the subregion with a fixed, user-defined field of view, that maximizes the integrated saliency within its gnomonic projection. A 2D Gaussian weighting function is applied to the predicted saliency values within each patch before integration to favor patches that center the most salient objects. While this is an intuitive approach, it is also an effective one. Results are shown in Figure 6.11 and,

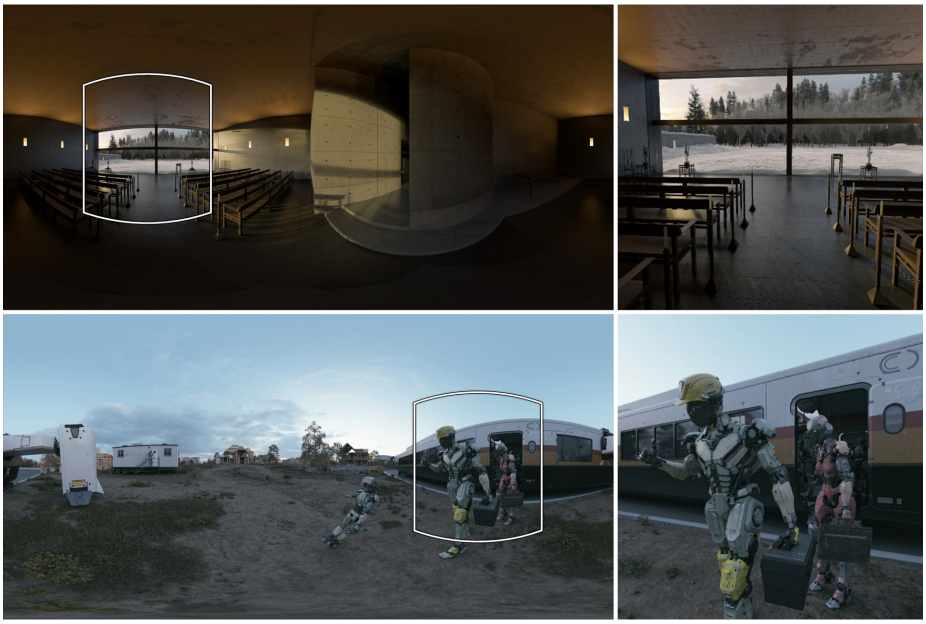


Figure 6.11: Automatic panorama thumbnail generation. The most salient regions of a panorama can be extracted to serve as a representative preview of the entire scene.

for all 22 scenes, in Appendix C. Note that this approach to thumbnail generation is also closely related to techniques for gaze-based photo cropping [274].

6.6.3 *Panorama video synopsis*

Automatically generating video synopses is an important and active area of research (e.g., [263]). Most recently, Su et al. [307, 308] introduced the problem of automatically extracting paths of a camera with a smaller field-of-view through 360° panorama videos, dubbed *pano2vid*. Good saliency prediction for monoscopic and stereoscopic VR videos can help improve these and many other applications. Figure 6.12, for example, shows an approach to combining video synopsis and *pano2vid*. Here, we predict the saliency for each frame in a video as discussed in Section 6.5.1, and extracted the panorama thumbnail from the first frame, as discussed in the previous subsection. In subsequent frames, we search for the window in the panorama with the highest saliency that is close to the center of the last window. Neither the saliency prediction step nor this simple search procedure enforce strict temporal consistency, but the resulting panorama video synopsis works quite well.

6.6.4 *Saliency-aware VR image compression*

Emerging VR image and video formats require substantially more bandwidth than conventional images and videos. Yet, low latency is even more critical in immersive environments than for desktop viewing scenarios. Thus, optimizing the bandwidth for VR video with advanced compression schemes is important and has become an active area of research [347]. Inspired by saliency-aware video compression schemes [127], we test an intuitive approach to saliency-aware compression for omni-directional stereo panoramas. Specifically, we propose to maintain a higher resolution in more salient regions of the panorama.

To evaluate potential benefits of saliency-aware panorama compression, we downsample a cube map representation of the omni-directional stereo panoramas with a bicubic filter by a factor of 6. We then upsample the low-resolution cube map and blend it with the 10% most salient regions of

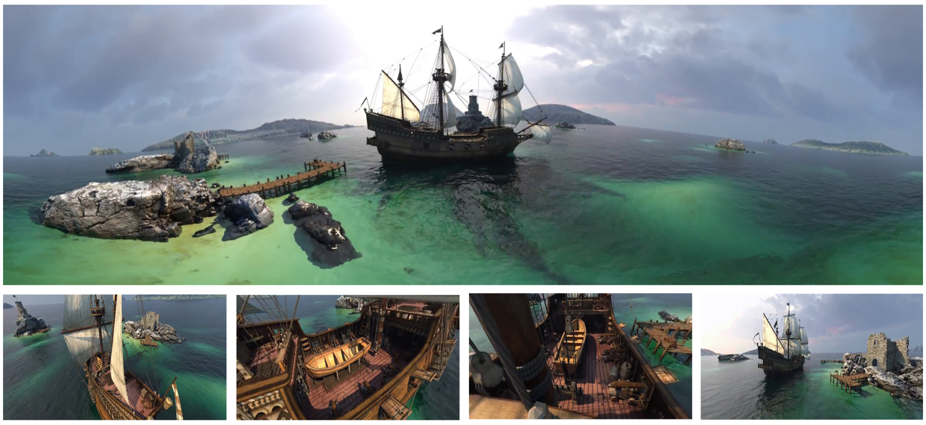


Figure 6.12: Automatic panorama video synopsis. Saliency prediction in VR videos can be used to create a short, stop-motion-like animation that summarizes the video. For this application, we predict saliency of each frame, extract a panorama thumbnail from one of the first video frames, and then search every N^{th} frame for the window with highest saliency within a certain neighborhood of the last window.

the high-resolution panoramas, using the ground-truth saliency maps. Overall, the compression ratio of the raw pixel count is thus 25%. Figure 6.13 shows this saliency-aware compression for an example image.

To evaluate the proposed saliency-aware VR image compression, we carried out a pilot study to assess the perceived quality of saliency-aware compression when compared to regular downsampling for a comparable compression ratio. To this end, users were presented with ten randomized pairs of stereo panoramas, and they were asked to pick the one that had better quality in a two-alternative forced choice (2AFC) test. For each pair, we sequentially displayed the two panoramas in randomized order, with a blank frame of 0.75 seconds between the two alternatives [249]. A total of eight users participated in the study, all reported normal or corrected-to-normal vision. The results of the study are shown in Figure 6.13 (bottom left). Saliency-aware compression was preferred for most scenes, and performed worse in only one scene. These preliminary results encourage future investigations of saliency-aware image and video compression for VR.

6.7 Discussion and conclusion

In summary, we collect a dataset that includes gaze and head orientation for users observing omni-directional stereo panoramas in VR, both in a standing and in a seated condition. We also capture users observing the same scenes in a desktop scenario, exploring monoscopic panoramas with mouse-based interaction. The data encompasses 169 users in three different conditions, totaling 1980 head and gaze trajectories.

The primary insights of our data analysis are: (1) gaze statistics and saliency in VR seem to be in good agreement with those of conventional displays; as a consequence, existing saliency predictors can be applied to VR using a few simple modifications described in this work; (2) head and gaze interaction are coupled in VR viewing conditions – we show that head orientation recorded by inertial sensors may be sufficient to predict saliency with reasonable accuracy without the need for costly eye trackers; (3) we can accurately predict time-dependent viewing behavior only within the first few seconds after being exposed to a new scene but not for longer periods of time due to the high inter-user variance; (4) the distribution of salient regions in the scene has a significant impact on how viewers explore a scene: the fewer salient regions, the faster user attention gets directed towards any of them and the more concentrated their attention is; (5) we

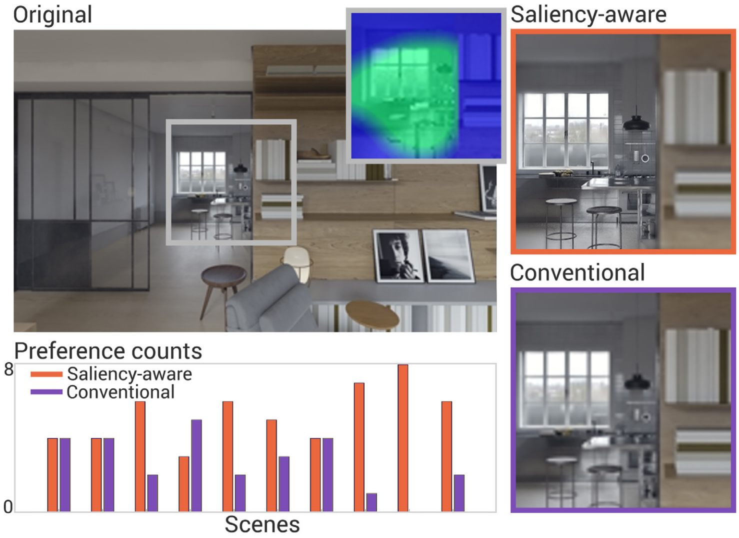


Figure 6.13: Saliency-aware panorama compression. *Top left:* original, high-resolution region of the input panorama. Inset shows the compression map based on saliency information, where green indicates more salient regions. *Right:* Close-ups showing the differences between saliency-aware compression and conventional downsampling. Note that salient regions retain a better quality in our compression, while non-salient regions get more degraded. *Bottom left:* Preference counts for the ten scenes displayed during the user study.

observe two distinct viewing modes: attention and re-orientation, potentially distinguishable via head or gaze movement in real time and thus useful for interactive applications.

These insights could have a direct impact on a range of common tasks in VR. We outline a number of applications, such as panorama thumbnail generation, panorama video synopsis, automatically placing cuts in VR video, and saliency-aware compression. These applications show the potential that saliency has for emerging VR systems and we hope to inspire further research in this domain.

FUTURE WORK Many potential avenues of future work exist. We did not use a 3D display or mobile device since we wanted to closely resemble the most ÒstandardÓ viewing condition (regular monitor or laptop). Alternative viewing devices could be interesting for future work. Nevertheless, one of our goals is to analyze whether viewing behavior using regular desktop screens is similar to using a HMD, and our analysis seems to support this hypothesis. We believe this is an important insight, since it could enable future work to collect large saliency datasets for omni-directional stereo panoramas without the need for HMDs equipped with eye trackers.

Predicting gaze scanpaths of observers when freely exploring a VR panorama would be very interesting in many fields, including vision, cognition, and of course, any VR-related application. Since the seminal work of Koch and Ulman [167], many researchers have proposed models of human gaze when viewing regular 2D images on conventional displays (e.g., [31, 126, 183, 334]). An important element to derive such models is gaze statistics, and whether those found in our VR setup are comparable to the ones reported for traditional viewing conditions; this would inform to what extent we can use existing gaze predictors in VR applications, or be useful as priors in the development of new predictors. Our data can be of particular interest to build gaze predictors using just head movement as input, since head position is much cheaper to obtain than actual gaze data.

Our data may still be insufficient to train robust data-driven behavioral models; we hope that making our scenes and code available will help gather more data for this purpose. We also hope it will be a basis for people to further explore other scenarios, such as dynamic or interactive scenes, the influence of the task, or the presence of motion parallax, etc. These future studies could leverage our methodology and metrics, and build upon them for the specific particularities of their scenarios. It would be interesting to explore how behavioral models could improve low-cost but imprecise gaze sensors, such as electrooculograms. Future work could also incorporate temporal consistency for saliency prediction in videos, or extend it to multimodal experiences that include audio.

7. Movie editing and cognitive event segmentation in virtual reality video

ABOUT THIS CHAPTER

The work presented in this chapter was presented at *SIGGRAPH* 2017, and published in *ACM Transactions on Graphics*. In this work we gather gaze data from viewers watching VR videos containing different edits with varying parameters, and provide the first systematic analysis of viewers' behavior and the perception of continuity in VR. While I led the line of work (under the supervision of Diego Gutierrez and Belen Masia), this work was conducted in collaboration with *Stanford University*. Additionally, as part of his final degree project supervised by myself, Jaime Ruiz-Borau collaborated setting up the experiments and conducting the user study.

A. Serrano, V. Sitzmann, J. Ruiz-Borau, G. Wetzstein, D. Gutierrez, and B. Masia.
MOVIE EDITING AND COGNITIVE EVENT SEGMENTATION IN VIRTUAL REALITY VIDEO.
ACM Transactions on Graphics, Vol. 36(4), 2017.

7.1 Introduction

Movies are made up of many different camera shots, usually taken at very different times and locations, separated by cuts. Given that the resulting flow of information is usually discontinuous in space, time, and action, while the real world is not, it is somewhat surprising that the result is perceived as a coherent sequence of events. The key to maintaining this illusion lies in how these shots are edited together, for which filmmakers rely on a system called *continuity editing* [32, 239]. Although other techniques exist to link shots together, such as the fade-out, fade-in, or dissolve, approximately 95% of editing boundaries are cuts [72], which directly splice two camera frames.

The goal of continuity editing in movies is then to create a sense of situational continuity (a sequence of shots perceived as a single event), or discontinuity (transitions from one event or episode to another). Professional editors have developed both a strong sense of rhythm and a solid intuition for editing together camera shots, making the cuts "invisible". For instance, spatial continuity is maintained largely by the 180° rule, stating that the camera should not cross the axis of action connecting two characters in a shot, while continuity of action is achieved by starting the action in one shot and immediately continuing it in the shot after the cut. Some authors have proposed partial theories to explain why these edited shots are perceived as a continuous event. For example, the 180° rule creates a virtual stage where the action unfolds [32], while mechanisms to process and track biological motion may mask an action cut [300].

However, the higher level cognitive processes that make continuity editing work are not yet completely understood. What is understood, though, is that a core component of spatial perception is our ability to segment a whole into parts [27]. Recent cognitive and neuroscience research indicates that a similar segmentation also occurs in the temporal domain, breaking up a continuous activity into a series of meaningful events. This has lead to the development of the *event segmentation* theory [177, 266, 351], which postulates that our brains use this discrete representation to predict the immediate course of events, and to create an internal, interconnected representation in memory. New events are registered whenever a change in action, space, or time, occurs. Based on this theory, recent works have explored how continuity is perceived in movies, across different types of cuts [73, 200, 352]. Interestingly, it seems that the predictive process suggested by event segmentation theory is consistent with common practice by professional movie editors.

In this work, we investigate continuity editing for *virtual reality videos*[18]. Virtual reality (VR) content is intrinsically different from traditional movies in that viewers now have partial control of the camera; while the position of the viewer within the scene is decided during acquisition, the orientation is not. This newly-gained freedom of users, however, renders many usual techniques, such as camera angles and zooms, ineffective when editing the movie. Nevertheless, new degrees of freedom for content creators are enabled, and fundamental questions as to what aspects of the well-established cinematographic language apply to VR should be revisited.In particular, we seek to investigate answers to the following key questions:

- Does continuity editing work in VR, i.e., is the perception of events in an edited VR movie similar to traditional cinematography?
- A common, safe belief when editing a VR movie is that the regions of interest should be aligned before and after the cut. Is this the only possible option? What are the consequences of introducing a misalignment across the cut boundaries?
- Are certain types of discontinuities (cuts) in VR favored over others? Do they affect viewer behavior differently?

We use a head mounted display (HMD), equipped with an eye tracker, and gather behavioral data of users viewing different VR videos containing different types of edits, and with varying parameters. We first perform a study to analyze whether the connections between traditional cinematography and cognitive event segmentation apply to immersive VR (Sec. 7.4). To this end, we replicate a recent cognitive experiment, previously carried out using traditional cinematographic footage [200], using instead a VR movie. Our results show similar trends in the data, suggesting that the same key cognitive mechanisms come into play, with an overall perception of continuity across edit boundaries.

We further analyze continuity editing for VR, exploring a large parameter space that includes the type of edit from the cognitive point of view of event segmentation, the number and position of regions of interest before and after the cut, or their relative alignment across the cut boundary (Sec. 4.4). We propose and leverage a series of novel metrics that allow to describe viewers' attentional behavior in VR (Sec. 2.4), including fixations on a region of interest, alteration of gaze after a cut, exploratory nature of the viewing experience, and a state sequence analysis of the temporal domain according to whether the viewer is fixating on a region or performing saccadic movements.

Our analyses reveal some findings that can be relevant for VR content creators and editors: for instance, that predictions from the cognitive event segmentation theory seem to be useful guides for VR editing; that different types of edits are equally well understood in terms of continuity; how the dependence of the time to convergence to a region of interest after a cut is not linear with the misalignment between regions of interest at the cut, but rather appears to follow an exponential trend; or how spatial misalignments between regions of interest at the edit boundaries elicit a more exploratory behavior even *after* viewers have fixated on a new region of interest. We believe our work is the first to empirically test the connections between continuity editing, cognition, and narrative VR, as well as to look into the problem of editing in VR, in a systematic manner. In addition, we provide all our eye tracking data, videos, and code to help other researchers build upon our work in our project website[19].

7.2 Related work

TOOLS FOR EDITING Creating a sequence of shots from raw footage while maintaining visual continuity is hard, especially for novice users [75]. Automatic cinematography for 3D environments was proposed by He at al. [132], encoding film *idioms* as hierarchical finite state machines,

[18] In this work we deal with 360° movies; throughout the text we will use the terms VR and 360° movies interchangeably.

[19] http://webdiis.unizar.es/~aserrano/projects/VR-cinematography

while Christianson et al. [64] proposed a declarative camera control system. Many other different tools have been devised to help in the editing process, usually leveraging the particular characteristics of specific domains such as 3D animations [108], interview videos [26], narrated videos [324], classroom lectures [133], group meetings [259], egocentric footage [198], or multiple social cameras [14]. Jain et al. [151] proposed a gaze-driven, re-editing system for retargeting video to different displays. More recently, Wu and Christie [345] created a language to define camera framing and shot sequencing. Other methods focusing on camera placement and planning can be found in the work of Christie et al. [65]. All these tools have been designed for traditional, two-dimensional viewing experiences, where the spectator sits passively in front of a screen. In contrast, our goal is to analyze continuity editing for virtual reality videos.

CONTINUITY AND COGNITION Several works have analyzed the effects of edits or cuts from a computer vision perspective (e.g., [49, 142, 300]). Closer to our approach, a few works have analyzed the perception of continuity from a cognitive science point of view. For instance, Cohn's analyses of comic strips [66] suggest that viewers can build links between frames while maintaining a global sense of the narrative; however, rearranging elements can quickly lead to confusion. Some researchers argue that our perception of reality is a very flexible process, and this flexibility allows us to adapt and perceive edited film as a continuous story [11, 72]. Smith [299] performed an empirical study to understand how continuity editing aligns with our perceptual abilities, identifying the role of visual attention in the perception of continuity between edits. In our work, we explore the recent theory of *event segmentation* [177, 266, 349, 351], and analyze its connections with continuity editing for VR.

7.3 Background on event segmentation

We present here a brief summary of the event segmentation theory, and refer the reader to the original publications for a more thorough explanation [177, 266, 349, 351]. Recent research suggests that event segmentation is an *automatic* key component of our perceptual processing, reducing a continuous flow of activity into a hierarchical, discrete set of events. The advantages of this strategy are twofold: First, it is very efficient in terms of internal representation and memory. Second, it provides a much easier way to think about events in relation to one another. It can be seen as the time equivalent to the well-known spatial segmentation in vision, where we segment an object (e.g., a car) into many components such as wheels, chassis, engine, etc.

This discrete mental representation is used as a basis for predicting the immediate course of events: a person walking down the street will continue to do so, or somebody will answer a question when asked. When these predictions are violated, it is an indication of a new (discrete) event; in other words, it seems that unexpected changes lead to the perception of an event boundary. More precisely, the event segmentation theory assumes that new events are registered when *changes in action, space, or time*, occur; when this happens, the mechanisms of event segmentation update the mental representation of the event, storing the old one in long-term memory.

This event segmentation theory has recently been tested in the context of film understanding. Some experiments have even recorded brain activity with functional magnetic resonance imaging (fMRI) while watching a movie, and showed that many regions in the cortex underwent substantial changes in response to the situational discontinuities (unexpected changes) introduced by some movie cuts [200, 352]. An interesting observation follows: *the predictive process suggested by event segmentation theory is consistent with common practice by professional movie editors*, who place cuts to support or break the expectations of event continuity by the viewers [32]. When a cut introduces a major change, the brain does not try to explain the perceived discontinuity; instead, it adapts to the change, creates a new mental representation, and begins populating it with details [200]. This automatic mechanism might be a key process to explain why continuity editing works. The next section explores whether this connection between event segmentation and continuity editing studied in traditional cinema carries over to VR movies, a key question before we can dive into a more detailed investigation. Note that, in the following, we use the term *edit* to refer to a discontinuity between two shots, while *cut* refers to the actual cinematographic implementation (match-on-action, jump cut, etc.) of the edit.

7.4 Does continuity editing work in VR?

As we have seen in Sec. 7.3, there is considerable evidence that continuity editing performed in traditional movies may be related to how our brains process events and situational changes, and that this may be the cause why continuity editing has been so successful in conveying the narrative. Therefore, before we analyze specific aspects related to editing in VR movies, we first want to assert that continuity editing applies to VR scenarios. For this purpose, we check whether the connections between event segmentation and edits, which have been identified and analyzed in traditional movies [200] also hold in VR movies, where the viewing conditions and the perception of immersion change significantly. This is the goal of the experiment described in this section. We aim to replicate the methodology of recent cognition studies, sharing a similar goal in the contexts of event segmentation [350], and film understanding [200, 352]. We introduce such works and our own experiment in the following paragraphs.

TYPES OF EDITS Following common practice in film editing, Magliano and Zacks [200] define a continuity domain along the dimensions of space, time, and action. They then classify edits into three different classes, which we call here E_1, E_2, and E_3:

- E_1: edits that are discontinuous in space or time, and discontinuous in action (action discontinuities);
- E_2: edits that are discontinuous in space or time, but continuous in action (spatial/temporal discontinuities);
- E_3: edits that are continuous in space, time, and action (continuity edits).

We adopt the same taxonomy for edits in this experiment, and in the rest of the chapter.

COGNITION STUDIES WITH TRADITIONAL MOVIE CONTENT In these studies [200, 352], participants watched *The Red Balloon* (a 33-minute, 1956 movie by A. Lamorisse), and were asked to segment the movie into meaningful events by pressing a button. They were asked to do this twice, once defining the *"largest units of natural and meaningful activity"* (coarse segmentation), and once defining the smallest units (fine segmentation); the order of this division was randomized between participants, who first practiced the task on a different, 2.5-minute movie. *The Red Balloon* was presented in 7-to-10-minute sections, to avoid fatigue. Previous to this task, the authors additionally identified all the locations where edits occurred in the movie, and coded each one according to the above categorization: E_1, E_2, or E_3. Based on the principles of film editing discussed in Sec. 7.3, action discontinuities E_1 should have the largest influence on perceived discontinuities, whereas continuity edits E_3 should mostly maintain the perceived continuity. The analysis of the data discretized in five-second bins, along with fMRI information, confirmed this predicted trend.

REPLICATION OF THE STUDY WITH VR CONTENT We followed the same methodology in our VR study. Specifically, we asked seven participants (ages between 21 and 31, three female) to watch *four* publicly available VR movies (see Appendix for details). Their initial task was to mark perceived event boundaries both at coarse and fine scales, similar to the original experiments. Previous to the experiment, the participants watched a training movie[20]. We did not find strong correlations between the fine event segmentation and the edits in the first tested VR movie. This is expected since, different from traditional movies, the time scale of such fine perceived events

[20] https://youtu.be/fz88kpRNTqM

is about one order of magnitude smaller than the average VR shot (seconds vs. tens of seconds). Therefore in the other three movies we only asked them to mark events at coarse scale; we analyze this data only in the rest of the chapter.

Participants watched the movies while seated, wearing an Oculus Rift. We chose these particular movies among several candidates since: i) like *The Red Balloon*, they are narrative movies with a rich enough structure; ii) they last less than eight minutes, which falls within the range fixed by previous studies to avoid fatigue; iii) their average shot lasts about 20 seconds, close to the average we obtained from analyzing several 360° movies; and iv) they contain edits of all three kinds. Figure 7.1 (top) shows representative frames of all three types of edits for one of these movies: *Star Wars - Hunting of the Fallen*.

INSIGHTS To investigate the relation between the edits and the perceived event segmentation, we identified the location and type of each edit, and binned all the event boundaries marked by the participants within a ±3 second window, centered at the edit. As expected, some perceived boundaries were not linked to edits, but to new events or actions within a shot (e.g., a new actor entering a scene, or the start of a conversation). Figure 7.1 (bottom) shows the results for all four movies, grouped by edit category. Our findings show similarities with the previous studies on traditional cinematography [200, 352]. First, action discontinuities dominate event segmentation, and are therefore the strongest predictors of event boundaries. Second, continuity edits succeed in maintaining a sense of continuity of action, even across the edit boundaries. It thus appears that the key cognitive aspects of traditional movie editing that make it work so well carry over to 360° immersive narrative movies. To our knowledge, this is the first time that this has been empirically tested.

7.5 Measuring continuity in VR

After confirming in the previous section that the perception of continuity is maintained across edit boundaries in VR narrative content, we now perform a second, in-depth study to assess *how the different parameters that define an edit in VR affect the viewers' behavior after the edit takes place*. Given the high dimensionality of this space, we focus on four main parameters (or *variables of influence*), which are: the type of edit, for which we follow the cognitive taxonomy described in previous sections; the degree of misalignment of the regions of interest (ROIs) before and after the edit; and the number and location of such ROIs, both before and after the edit boundaries. In the following we describe our stimuli, variables of influence, and procedure. Additional details can be found in the Appendix, while a complete table with all the possible combinations of conditions tested can be found in Appendix D.

7.5.1 *Stimuli*

Our stimuli are created from 360° monocular videos, professionally captured and stitched by a local company. We choose to use monocular (and not stereo) footage since it is more common among existing capture devices and public repositories (e.g., YouTube 360). The videos depict four different scenarios (*Stairs, Kitchen, Living Room, Study*), with four different actions in each one, totalling sixteen videos ranging from 13 seconds to 2 minutes in length. Figure 7.2 shows some representative frames in equirectangular projection. They were captured using two different rigs: a GoPro Omni (a 360-video rig consisting of six GoPro Hero4 cameras), and a Freedom360 3× rig (with three GoPro Hero4 cameras with modified Entaniya 220 lenses). Sound was recorded using a Zoom F8 recorder with wireless microphones.

From these videos, we created a total of 216 *clips*, sampling our parameter space as explained in the following subsection. Each clip is made up of two shots, separated by an edit. Shots are taken from short sequences both within and across the four different scenarios, to maximize variety.

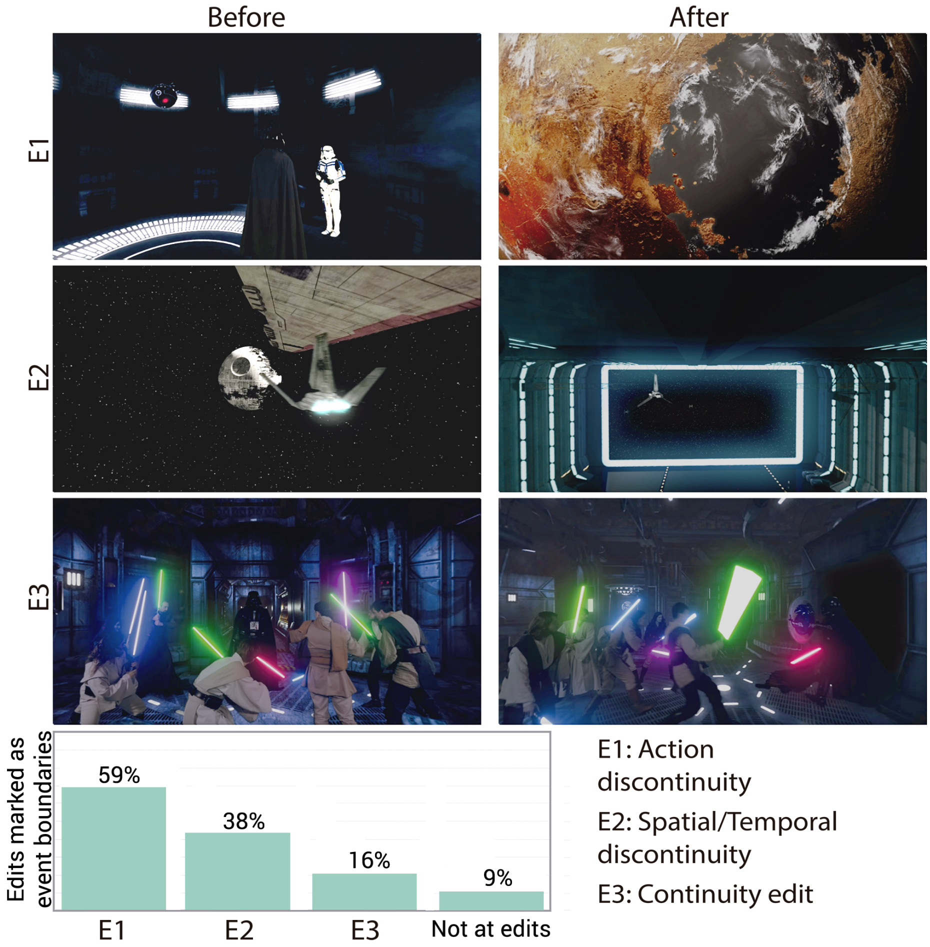


Figure 7.1: **Top:** Representative frames of the 360° movie *Star Wars - Hunting of the Fallen* (original video property of CubeFX; video and images used with permission), before (left) and after (right) an edit for each of the three types of edits. Top row: action discontinuity (E_1). The frame before the edit is not related with the frame after the edit; there is a complete change of action, space, and time. Middle row: spatial discontinuity (E_2). The edit follows the action by the spaceship (match-on-action), but changes the location; the action is the element connecting the frames before and after the edit. Bottom row: continuity edit (E_3). The same scene is depicted before and after the edit with a continuity in space, time, and action; only the camera angle changes. **Bottom:** Results of our coarse segmentation test, showing the percentage of edits of each type marked as an event boundary by subjects, and normalized by the number of occurrences of each type of edit. E_1 action discontinuities dominate event segmentation, while E_3 continuity edits maintain the perceived continuity of the event. There is also a small percentage of event boundaries that were marked not at edits. These findings match results of similar studies in traditional cinematography, and suggest that common practice can be in general applied to narrative VR as well.

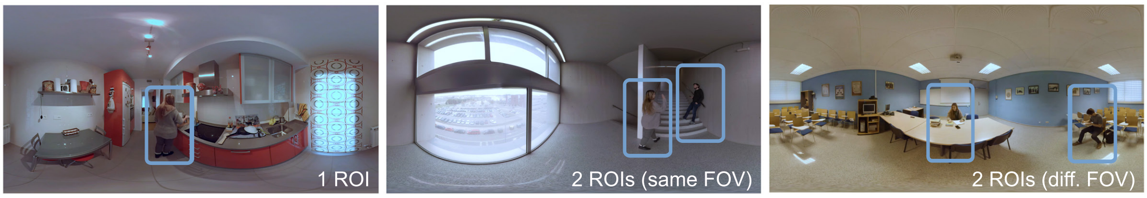


Figure 7.2: Representative frames of three of the scenes depicted in our clips: *Kitchen*, *Stairs*, and *Study* (refer to the project website for full videos). From left to right, examples corresponding to the following region of interest (ROI) configurations: 1 ROI, 2 ROIs in the same FOV, and 2 ROIs in different FOV. For clarity, ROIs are marked by a blue box.

Each shot lasts six seconds, to provide enough time to the viewers to understand the actions being shown.

7.5.2 *Variables of influence and parameter space*

TYPE OF EDIT We rely on the event segmentation theory, and initially consider the three different types of edits $\{E_1, E_2, E_3\}$, defined along the dimensions of space, time, and action, as introduced in Sec. 7.3. However, after analyzing seventeen VR movies we have observed that E_3 (which essentially refers to a change of viewpoint within the same scene) is rarely used in narrative VR. In these movies, 73% of the edits corresponded to E_1, 25% to E_2, and only 2% to E_3. This differs from traditional movies, where most of the edits are continuity edits (E_3) [200], and reflects an interesting contrast between the established storytelling techniques for the two media. Due to the rare appearance of E_3 edits in VR movies, we remove it from our conditions, and focus on the two most prominent types of edits: $\mathbf{E} = \{E_1, E_2\}$.

For the actual implementation of these edits, we revise traditional cinematography techniques and analyze existing VR videos, and select the most common cuts for each type of edit: For type E_1 (discontinuous in action, and in time or space) we use jump cuts, while for E_2 (continuous in action, discontinuous in time *or* space) we use compressed time cuts, and match-on-action cuts (see, e.g., [52, 83]; please refer to the Appendix for a brief explanation of each one). To keep a balanced number of clips for each type of edit, we include twice as many jump cuts (type E_1) as match-on-action and compressed time cuts (type E_2).

ALIGNMENT OF ROIS We define the regions of interest (ROIs) as the areas in the 360° frame in which the action takes place[21]. Since the point of view of the camera cannot be controlled by the filmmaker in VR, a common practice among content creators is to simply align ROIs before and after an edit, to make sure that the viewer does not miss important information. However, the exploration of controlled (mis)alignments is interesting for the following reasons: First, the director may want to introduce some misalignment between ROIs for artistic or narrative purposes (e.g., to create tension). Second, the viewer may not be looking at the predicted ROI before the cut, thus rendering the alignment after the edit useless. Third, there might be multiple ROIs within a scene. We therefore test three different alignment conditions: (i) perfect alignment before and after the edit (i.e., 0° between ROIs); (ii) a misalignment that is just within the field of view (FOV) of the HMD[22]; we chose 40° since it is close to the average misalignment in 360° videos found in public repositories; and (iii) a misalignment that is outside the FOV; we chose 80°, since we found that larger values are very rare. We name these conditions $\mathbf{A} = \{A_0, A_{40}, A_{80}\}$.

[21] We manually label ROIs as continuous regions at several keyframes, creating the rest through interpolation. We define the center of each ROI as the centroid of its pixels.

[22] Our Oculus DK2 HMD has a horizontal FOV of 95°, so a 40° misalignment falls just in the periphery of the FOV.

ROI CONFIGURATION The control of the viewer over the camera also makes the disposition and number of ROIs in the scene play a key role in gaze behavior. To analyze the ROI configuration before and after the edit, we introduce two variables, $\mathbf{R_b}$ and $\mathbf{R_a}$ respectively. The space of possible configurations is infinite, so to keep the task tractable we test three possibilities for each one: a single ROI ($R_{\{b|a\},0}$), two ROIs both falling within a single FOV ($R_{\{b|a\},1}$), and two ROIs not within the same FOV, i.e., more than 95° apart ($R_{\{b|a\},2}$). Examples of the three configurations are shown in Figure 7.2. The possible combinations of $\mathbf{R_b}$ and $\mathbf{R_a}$ yield a total of nine conditions.

SUMMARY This sampling leads to 2 types of edit, 3 alignments, and 9 ROI configurations, totaling 54 different conditions. For each one, we include four different clips, to minimize the effect of the particular scene shown, yielding our final number of 216 stimuli.

7.5.3 *Hardware and procedure*

We used an Oculus DK2 HMD equipped with a binocular eye tracker from pupil-labs[23], which records data at 120 Hz with a spatial accuracy of 1 degree. We also used a pair of headphones to reproduce stereo sound. Subjects stood up while viewing the video. A total of 49 subjects (34 male, 15 female, $\mu_{\text{age}} = 25.4$ years, $\sigma_{\text{age}} = 7.7$ years) participated in the experiment. All of them reported normal or corrected-to-normal vision. Each subject first carried out the eye tracker calibration procedure. Then, they were shown 36 stimuli from the total of 216, in random order. This randomization was such that no subject viewed two alignment conditions of the same clip, while guaranteeing that each clip was viewed by at least five people. It also avoids potential learning and fatigue effects affecting the results. Following Sitzmann et al. [298], in order to ensure that the starting condition was the same for all subjects, a gray environment with a small red box was displayed between clips; users had to find it and align their head direction with it, which would launch a new clip after 500 ms. The Unity game engine was used to show the videos, and to record head orientation on the same computer, while eye tracking data was recorded on a second computer. After viewing the clips the experimenter did a debriefing session with the subject. The total time per experiment was around 15 minutes. From the raw gathered data, we performed outlier rejection and then computed *scanpaths*, defined as a temporal sequence containing one gaze sample per frame. More details on these aspects can be found in the Appendix (gaze data processing and outlier rejection), and in Appendix D (data collection and debriefing). From this data we define, compute, and analyze a series of metrics, as described in the next section.

7.6 How do edits in VR affect gaze behavior?

To obtain meaningful data about viewers' gaze behavior across event boundaries in VR, we first gather additional *baseline data* to compare against. We make the assumption that the higher the gaze similarity between the edited clips and the corresponding (unedited) baseline videos, the higher the perception of continuity; this assumption is similar to previous works analyzing gaze to assess the impact of retargeting and editing operations in images and video [50, 151]. In the following, we first describe how this baseline data is obtained, then introduce our continuity metrics, and describe the results of our analysis.

7.6.1 *Baseline data*

In order to capture baseline eye tracking data from the unedited videos, we gathered ten new subjects (nine male, one female, $\mu_{\text{age}} = 28.1$ years, $\sigma_{\text{age}}= 5.2$ years) and collected head orientation and gaze data following the procedure described in Sec. 7.5.3. Videos were watched in random order. We compute the baseline scanpaths for each video from the obtained gaze data as the

[23] https://pupil-labs.com/

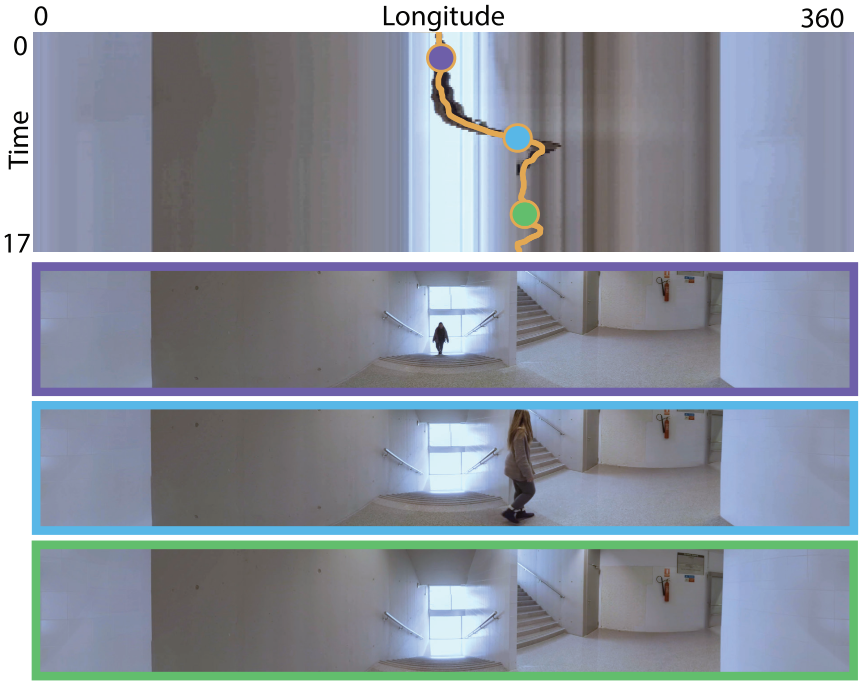


Figure 7.3: Subjects mean scanpath for one example video. We plot one scanline per frame of the video, the x-axis showing longitudinal position (0°-360°), and the y-axis time. Superimposed (orange line) we plot the scanpath, showing the temporal evolution of the longitudinal position of the gaze. We also plot the full frame (equirectangular projection) at three key instants that correspond to the three marked temporal instants. Viewers' gaze is clearly directed by the movement of the ROI along time.

mean scanpath across users. We show in Figure 7.3 the mean scanpath corresponding to one of our videos: We display the temporal evolution of the longitudinal gaze position (0°- 360°), and it shows how viewers' attention is driven towards the ROI moving across the scene.

To ensure that this data can be used as baseline for our subsequent analyses, we need to ascertain the congruency between subjects. To do so, we rely on a *receiver operating characteristic* curve (ROC) metric, which provides a measure of the *Inter Observer Congruency* (IOC) [182] over time. First, we aggregate all the users' fixations (please refer to the Appendix for a description of how fixations are computed from gaze data) in two-second windows, and convolve them with a 2D Gaussian of $\sigma = 1$ degree of visual angle [181], yielding a saliency map for each time window. The corresponding ROC curve is then obtained using a one-against-all approach by leaving out the i_{th} subject: we compute, for each saliency map, the k% most salient regions, and then calculate the percentage of fixations of the i_{th} subject that fall within those regions. This process is performed for a set of thresholds $k = 0\%..100\%$, and the resulting points define each curve. Additionally, we compute the *Area Under the Curve* (AUC) for each window, which provides an easier interpretation of the evolution of the IOC along time (Figure 7.4, right). The AUC takes values between 0 (incongruity between users) and 100 (complete congruency). As displayed in Figure 7.4, the congruency between subjects remains very high along time. On the left of the figure, the IOC rapidly reaches a value of 1 with $k = 2\%$ most salient regions, and remains constant for increasing values of k. On the right, the same interpretation from an AUC perspective: all the viewer's fixations fall on average within the 2% regions considered most salient by the rest of the viewers, yielding a very high AUC. This indicates that all the viewers consistently considered the same regions salient. Please, refer to the project website for the results for all our videos.

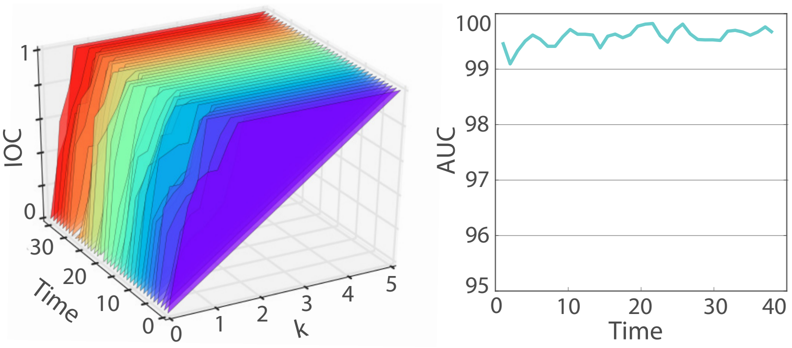


Figure 7.4: **Left**: Inter Observer Congruency (IOC) for one of our videos. We compute a ROC curve for each second of the video. **Right**: Temporal evolution of the Area Under the Curve (AUC) calculated for each of the ROC curves. The high values of the IOC and AUC indices indicate that all the viewers consistently considered the same regions salient (refer to the main text for details).

7.6.2 *Metrics*

Measuring the perceived continuity across edit boundaries in an objective manner is not a simple task, since no predefined metrics exist. We describe here our four different metrics used to analyze gaze behavior after an edit. In addition, to further look for underlying patterns in the users' behavior that our metrics may not capture, we also introduce a *state sequence* analysis.

frames to reach a roi (*framestoroi*) This is the simplest of our metrics, simply indicating the number of frames after the occurrence of the edit before the observer fixated on a ROI. It is indicative of the time taken to converge again to the main action(s) after the edit.

percentage of total fixations inside the roi (*percfixinside*) This percentage is computed after fixating on a ROI after the edit. It is thus independent of *framesToROI*. Different configurations of the ROIs may imply, by nature, different number of fixations inside the ROI. To compensate for this, we compute *percFixInside* relative to the average percentage of fixations inside a ROI, for each ROI configuration, *before* the edit. This metric is indicative of the interest of the viewer in the ROI(s).

scanpath error (*scanpatherror*) We compute the RMSE of each scanpath with respect to the corresponding baseline scanpath (see Sec. 7.6.1). This metric indicates how gaze behavior is altered by the edit; again, we compute this metric after fixating on a ROI after the edit, to make it independent of *framesToROI*.

number of fixations (*nfix*) We compute the ratio between the number of fixations, and the total number of gaze samples after the edit after fixating on a ROI; this way, we eliminate the possible increase in saccades while searching for the ROI after the edit. This metric is therefore indicative of how many fixations and saccades the subject performs. A low value corresponds to a higher quantity of saccades, which in turn suggests a more exploratory behavior, fixating less on any particular region or action.

state sequences We classify users' fixations along time in four different states, corresponding to the ROIs (each clip having one, or two), the background, and a so-called idle state where saccadic eye movements take place and no fixations are recorded. With this classification we are able to describe users' behavior as a state sequence, observing the succession of states with time, as well as the time spent in each of them. In particular, we use a *state distribution analysis* to represent the general pattern of state sequences for each condition, which provides an aggregated view of the frequency of each state for each time interval. We use the R library TraMineR [106] for this analysis.

7.6.3 *Analysis*

Since we cannot assume that our observations are independent, we employ multilevel modeling [39, 262] in our analysis, which is well-suited for grouped or related data like ours. Multilevel modeling allows the specification of random effects among the predictors, i.e., it contemplates the possibility that the model might differ for different values of these random effects. In our case, the random effect is the particular subject viewing the stimuli, for which we considered a random intercept.

We include in the regression all four factors ($\mathbf{A}$, $\mathbf{E}$, $\mathbf{R_b}$ and $\mathbf{R_a}$), as well as the first-order interactions between them. Since we have categorical variables among our predictors, we recode them to dummy binary variables for the regression. For two of our metrics (*percFixInside* and *nFix*), the effect of the subject was significant ($p = 0.002$ and $p = 0.005$, respectively, in Wald's test), indicating that we cannot treat the samples as independent; we therefore report significance values given by multilevel modeling. For the other two metrics (*framesToROI* and *scanpathError*), the effect of the subject was found to be non-significant ($p = 0.201$ and $p = 0.046$, respectively). Therefore, samples can be considered independent, and we perform factorial ANOVA, together with Bonferroni post hoc analyses to further look for significant effects in our data. Throughout the analysis we use a significance level of 0.01.

INFLUENCE OF PREVIOUS VR EXPERIENCE In addition to analyzing the influence of the different factors, detailed below, we also analyze whether the subjects' previous experience using VR had an effect on the results. We record this information in the pre-test questionnaire. None of our subjects used VR frequently, but 69% of them had used VR before at some point. When looking at the effect of this previous VR experience on the metrics employed, we found that it had no effect on any of the metrics tested ($p = 0.600$ for *percFixInside*, $p = 0.832$ for *nFix*, $p = 0.197$ for *framesToROI*, and $p = 0.480$ for *scanpathError*). This is to be expected, since an occasional or rare use of VR is unlikely to cause any change in the results.

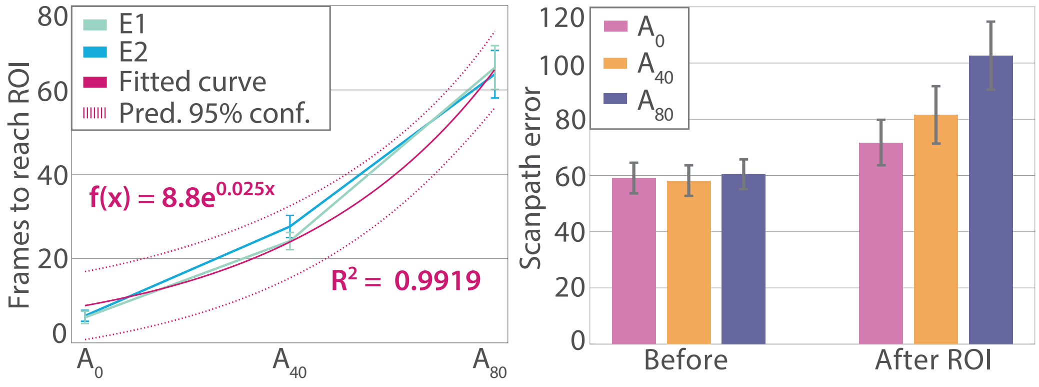


Figure 7.5: **Left:** Average *framesToROI* for each alignment. The green and blue curves show average data for the two types of edit (E_1 and E_2, respectively). We also show a fit to an exponential function, with the associated 95% confidence interval. **Right:** Mean RMSE with respect to the baseline before the edit, and after the edit after seeing the ROI (*scanpathError*) for the different alignment conditions tested. In both plots, error bars show a 95% confidence interval for the mean.

INFLUENCE OF ALIGNMENT **a** The first thing we observe is that there is a clear effect of the alignment factor on the four dependent variables (metrics) under study. In the case of the *framesToROI* ($F(2, 787) = 198.059, p < 0.001$), the Bonferroni post hoc further shows a significant difference ($p < 0.001$) between all three levels (A_0, A_{40} and A_{80}). As expected, the further away the ROI is, the longer it takes viewers to find it. Interestingly, the metric suggests an exponential trend with the degrees of misalignment. This is shown in Figure 7.5 (left), which includes the goodness of fit, and the 95% confidence interval. Figure 7.6 also illustrates this, with strong peaks and larger tails of background fixations after the edit (t = 6 secs.) for A_{80} (bottom row) than A_0 (top row).

Our *scanpathError* metric ($F(2,787) = 14.511, p < 0.001$) allows us to dig deeper into this finding, showing in the post hoc analyses that there is no significant difference between A_0 and A_{40} ($p = 0.277$), while A_{80} is significantly different to both of them ($p \leq 0.001$ in both cases). This is shown in Figure 7.5 (right), comparing directly with the equivalent values before the edit (where, as expected, no significant difference was found). A similar trend can be seen in *percFixInside*: A_{80} is significantly different to A_0 ($p < 0.001$), but A_{40} is not ($p = 0.138$).

This effect seems to indicate that the large misalignment alters viewer behavior not only in the time it takes to fixate on the ROI, but also *after* it is found. A closer look reveals that the same significant difference holds for *nFix*: the number of fixations is significantly lower ($p = 0.003$) for A_{80} compared to A_0, but this is not the case for A_{40} ($p = 0.954$). This, also shown in Figure 7.7 (top) as a radar plot, is a very interesting finding, suggesting that viewers could be more inclined to explore the scene when there is a high misalignment across the edit boundary.

INFLUENCE OF TYPE OF EDIT **e** Interestingly, the type of edit (E) has no effect on the fixational behavior after the edit after fixating on the ROI ($p = 0.674$ and $p = 0.430$ for *percFixInside* and *nFix*, respectively). The type of edit did not have a significant effect on *scanpathError* ($F(1,787) = 0.038, p = 0.846$) either, but the interactions of the type of edit with both ROI configurations did ($p = 0.002$ in both cases). Surprisingly, the type of edit had no significant effect on the *framesToROI* either ($F(1,787) = 1.373, p = 0.242$), as hinted in Figure 7.5.

INFLUENCE OF ROI CONFIGURATIONS $\mathbf{r_b}$ AND $\mathbf{r_a}$ We observe no significant influence of these factors on *nFix*, indicating that ROI configuration does not influence the exploratory behavior (how much viewers fixate, in general) of the viewers after the edit once they see one of the ROIs. Interestingly, however, $\mathbf{R_b}$ has an effect on *percFixInside*, i.e., on how much viewers fixate on the ROI(s) after the edit after fixating, compared to the total number of fixations in that time period. Note that, while different ROI configurations may imply by nature different number of fixations inside, we are compensating for this effect in the computation of *percFixInside* (Sec. 7.6.2). Specifically, $\mathbf{R_b}$ reveals a difference between two ROIs in the same FOV, and one ROI ($R_{b,1}$ vs. $R_{b,0}$, $p = 0.015$), but not in case of two ROIs in different FOVs ($R_{b,2}$ vs. $R_{b,0}$, $p = 0.792$). This can be seen in Figure 7.7 (middle): two ROIs in the same FOV before the edit lead to less fixations on the ROI(s) after the edit. We hypothesize that this is because multiple ROIs before the edit elicit a more exploratory behavior after the edit, in search for more ROI(s) even after having fixated on one.

We also found a significant influence of the ROI configuration after the edit $\mathbf{R_a}$ on the deviation of the scanpath wrt. the baseline, *scanpathError* ($F(2,787) = 168.569, p < 0.001$ for $\mathbf{R_a}$); meanwhile, $\mathbf{R_b}$ had no significant influence ($F(2,787) = 1.660, p = 0.191$ for $\mathbf{R_b}$). Bonferroni post hocs show that $R_{a,2}$ is significantly different to the other two ($p < 0.001$), while $R_{a,0}$ and $R_{a,1}$ are not significantly different between them ($p = 0.804$). Figure 7.7 (bottom) shows this effect: the *scanpathError* is significantly higher for $R_{a,2}$ (two ROIs in different FOVs), indicating that there is more variability in the scanpaths since the two ROIs cannot be looked at simultaneously. Finally, both $\mathbf{R_b}$ and $\mathbf{R_a}$ had also a significant effect on *framesToROI* ($F(2,787) = 6.478, p = 0.002$ for $\mathbf{R_b}$, and $F(2,787) = 10.300, p < 0.001$ for $\mathbf{R_a}$).

OTHER EFFECTS Additionally, we can observe some new effects in the state distribution sequences (Figs. 7 and 9). In particular, we find an *exploration peak* right at the beginning of each clip, both when the video starts and right after the edit; this peak usually lasts around 1-2 seconds. It is followed by an *attention peak*, again lasting around 1-2 seconds. This effect appears regardless of the ROI configurations and the alignment, and can be observed in Figure 7.6. This suggests that users require some time to understand their environment and stabilize their gaze patterns when a change of scenario occurs; after that transitory state, however, their gaze is strongly attracted to the actions being performed (the ROIs).

Last, we analyze more in depth the effect of the two types of edits (E_1 and E_2) in the particular case of ($R_{b,0}$,$R_{a,0}$) (edits from one ROI to one ROI). This is one of the simplest cases, but also one of the most relevant, since many current VR film-making strategies are commonly based on a single ROI across scenes. In Figure 7.8 we show the *state distribution* for this particular case ($R_{b,0}$,$R_{a,0}$) for alignments A_0 and A_{80}, and for the two types of edits. Even though we found no significant effect of the type of edit in our metrics, the graphs suggest a difference that our metrics are not capturing. In particular, it seems that E_2 attracts more attention to the ROI after the edit than E_1, as seen in the deeper blue valley after the edit in the right column), and this effect is consistent across all alignments. A potential explanation is that the continuity in action before and after the E_2 edit acts as an anchor.

7.7 Discussion and conclusions

To our knowledge, our work is the first to attempt a systematic analysis of viewer behavior and perceived continuity in narrative VR content. A systematic exploration of this topic is challenging for two main reasons: (i) the extreme high dimensionality of its parameter space; and (ii) that it involves many discrete, categorical (as opposed to interval or ordinal) variables of influence. Moreover, other basic issues need to be addressed, such as: How does one measure continuity, or viewer behavior? Which are the best metrics to use? Are our observations independent of the subjects? We have relied on the event segmentation theory, which has provided us with some solid ground to carry out our research, and have analyzed previous related studies on traditional cinematography.

Our results may have direct implications in VR, informing content creators about the potential responses that certain edit configurations may elicit in the audience. For instance, for a fast-paced action movie our results suggest that ROIs should be aligned across edits, while to evoke a more exploratory behavior, misalignments are recommended. Additionally, from all the narrative 360° movies we have explored, we have found an interesting trend in the number and classification of edits: while in VR movies the great majority of edits are type E_1 (action discontinuity), they are by far the least frequent in traditional cinematography, where E_3 continuity edits are the most prominent. For example, *The Red Balloon* has 85 continuity edits, 67 spatial/temporal discontinuities, and only 18 action discontinuities. We believe this is due to the immersive nature of narrative VR, where an excessive number of continuity edits would reduce opportunities for free exploration. In the rest of the section, we summarize our main findings, and outline interesting areas of future work ahead.

COGNITION AND EVENT SEGMENTATION IN VR We have first replicated an existing cognitive study carried out on the *The Red Balloon* movie, and found many similarities in VR. Like in traditional cinematography, action discontinuities dominate event segmentation in VR, becoming the strongest predictors of event boundaries. Continuity edits do succeed in maintaining the perceived continuity also in VR, despite the visual discontinuity across edit boundaries. This suggests that viewers build a mental model of the shown event structure that is similar to watching a traditional movie, despite the drastically different viewing conditions.

MEASURING CONTINUITY EFFECTS Our analysis has revealed several other interesting findings. Moreover, most of our reported findings have significant values of $p < 0.01$; this minimizes the risk of false positives in our conclusions.

The relation between how misaligned a ROI appears after an edit, and how long it takes viewers to fixate on it, seems to be exponential; this could be used as a rough guideline when performing edits. Even more importantly, large misalignments across edit boundaries do alter the viewers' behavior *even after they have fixated on the new ROI*. A possible interpretation is that the misalignment fosters a more exploratory behavior, and thus could be used to control attention. Two ROIs in the same FOV before an edit seem to elicit a more exploratory behavior as well, even after having located one ROI after the edit.

Other effects not caught by our metrics can be inferred by visual inspection of the state distributions. There seems to be at exploration peak at the beginning of each clip, and a similar attention peak right after the edit, independent of the type of edit. Both suggest that users require some time to adapt to new visual content, before their gaze fixates on ROIs. Also, it appears that the ROI attracts more attention after an E_2 edit than after a type E_1, perhaps because the consistent action before and after the edit acts as an anchor.

LIMITATIONS AND FUTURE WORK As in all studies of similar nature, our results are only strictly valid for our chosen stimuli. We have focused on short 360° videos for several reasons: to isolate simple actions, avoiding confounding factors; to gain control over the stimuli, enabling a systematic exploration of the parameter space; and to facilitate the analysis of the gathered data. Some of our findings may therefore not generalize to conditions outside our study.

Of course, many other variables and parameters can be explored in future work, such as other types of cinematographic cuts, longer movies, more complex visual content, the influence of sound, or the effect of fatigue or frequent exposure to VR content. More comprehensive subjective data may also be a valuable source of information, together with our objective gaze data. We believe that the joint study of cognitive mechanisms and cinematographic techniques provides a solid ground to carry out this research.

In summary, we believe that our work is a timely effort, since VR videos are a fast-growing new medium still in its initial exploratory phase, with many content creators testing ways to communicate stories through it. We hope that our findings will be useful as guidelines for VR content creators, especially amateurs, across a reasonable range of situations.

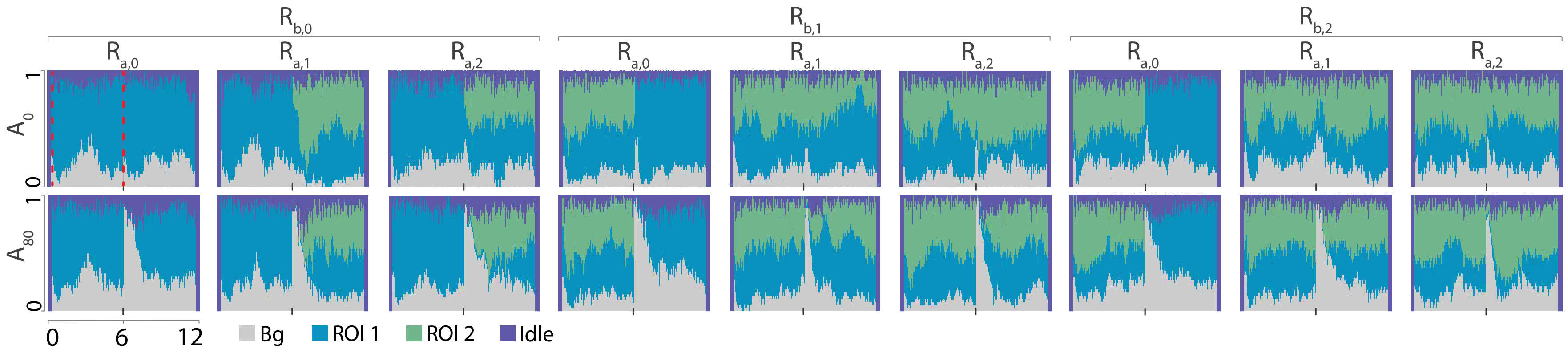


Figure 7.6: *State distribution* for all the different combinations of $\mathbf{R_b}$ and $\mathbf{R_a}$, and for alignments A_0 (first row) and A_{80} (second row). The different types of edits E are aggregated in each of the aforementioned conditions. The abscissae show time in seconds (the edit takes place at $t = 6$), while the ordinates show aggregated percentage of users. Each plot shows the percentage of users in each state at each time instant. The short Idle periods at the beginning and the end are due to black frames before and after the movie, and thus are not significant to our analysis. The two red lines on the top-left image illustrate the exploration and attention peaks, respectively, reported in this work.

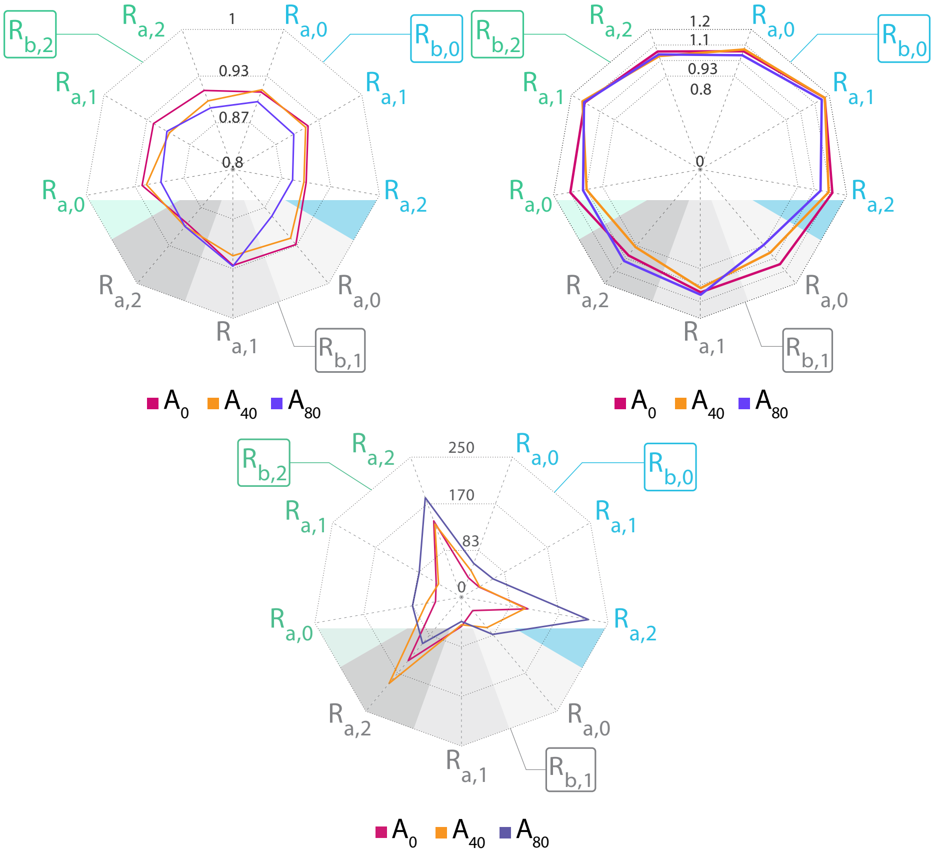


Figure 7.7: Radar graphs showing variation of three of our metrics with **A**, $\mathbf{R_b}$ and $\mathbf{R_a}$. Variation with **E** is not shown. In each graph, the three curves correspond to the three alignment conditions, as labeled in the legend. The radii of the graph correspond to the different combinations between R_a and R_b. R_a values are written at each point in the perimeter, while the large colored sectors (blue, gray and green), correspond to $\mathrm{R_b}$ ($\mathrm{R_{b,0}}$, $\mathrm{R_{b,1}}$ and $\mathrm{R_{b,2}}$ respectively, as indicated). **Left:** Number of fixations after the edit after fixating on the ROI (*nFix*), which is significantly different for A_{80} than for A_{40} and A_0. The scale of the radial axis is enlarged for visualization purposes. **Center:** Value of *percFixInside* for the different conditions; *percFixInside* is significantly affected by the ROI configuration both before and after the edit (see text for details). **Right:** Mean RMSE with respect to the baseline after the edit after fixating on the ROI (*scanpathError*). Please see the text for details.

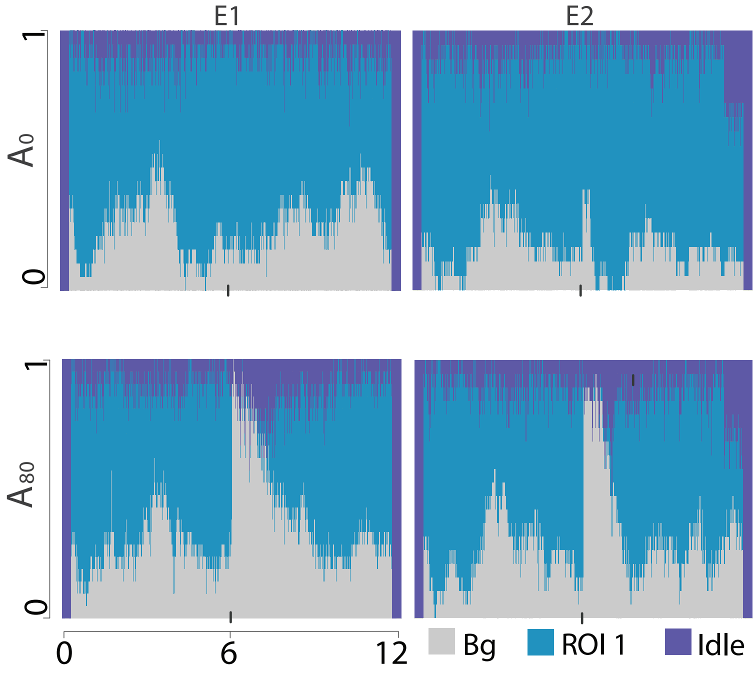


Figure 7.8: *State distribution* for $(R_{b,0}, R_{a,0})$, for alignments A_0 and A_{80}, and for the two different types of edits E_1 (left) and E_2 (right). Although our metrics did not capture this effect, it appears that E_1 edits might be harder to understand than E_2, as indicated by the deeper blue valley after the edit in the right column.

8. Motion parallax for 360 RGBD video

The work presented in this chapter was presented at the *IEEE VR* 2019 conference, and is to be published in *IEEE Transactions on Visualization and Computer Graphics*. In this work we propose a method for adding motion parallax (and thus six degrees of freedom) to 360° videos. This research was conducted in collaboration with *Adobe Research*, where I worked as an intern during the first stages of this project.

A. Serrano, I. Kim, Z. Chen, S. DiVerdi, D. Gutierrez, A. Hertzmann, and B. Masia.
Motion parallax for 360 RGBD video
IEEE Transactions on Visualization and Computer Graphics, 2019. To appear.

8.1 Introduction

With the growth of Virtual Reality (VR) headsets, stereo 360 video is becoming increasingly popular. Existing devices, including the GoPro Odyssey, Yi Halo, Vuze VR, Jaunt ONE or Facebook Surround 360, capture video that is converted to a stereo 360° video format for viewing in headsets. This video creates a feeling of "immersion" because it fills the viewer's field of view. However, this representation does not support motion parallax when the viewer shifts their head translationally. This experience is unnatural, can break the sense of immersion, and can cause discomfort and even nausea in some users, due to the mismatch between the visual and vestibular systems [317]. Even if the viewer attempts to stay static and carry out only rotational movements, accidental motion is likely to happen, leading to potential discomfort and sickness. Viewers that can display imagery with rotation and translation are commonly referred to as *6-DoF*; correspondingly, rotation-only viewing is referred to as *3-DoF*.

HMD-ready demos exhibiting motion parallax in synthetic and real world scenarios, such as *Welcome to Light Fields* [242], have shown the potential of 6-DoF viewing. Experiments have been carried out comparing 3-DoF and 6-DoF viewing, showing the importance of motion parallax to the viewing experience in VR, and how 6-DoF viewing leads to higher immersion and realism, and lower discomfort [275, 317]. At present, however, there is no practical system for capturing general-purpose 6-DoF video. While a number of research prototypes have been demonstrated, most of them only work for static scenes (e.g., [135, 199]), require impractical amounts of video storage for a reasonable range of head motion [279], require complex indoor setups and only capture actors within a constrained volume [363], or, else are still only proofs-of-concept that—as well as can be determined from available information—seem to require large, expensive setups [149].

We introduce a new approach to adding motion parallax to footage recorded by existing 360° video capture systems. We first obtain a suitable depth map from the input video. An initial depth can be provided by existing 360 stitching algorithms (e.g., [13]), or we can estimate it using an off-the-shelf deep learning algorithm [112]. A baseline approach would be to directly create a 3D scene from this initial depth map. However, because conventional depth map algorithms are not designed for reprojection, the baseline approach creates objectionable artifacts (Figure 8.1), including lack of disocclusions, jagged boundaries, and lack of temporal coherence. Our approach is as follows. We introduce a three-layer representation of the video for playback, which allows disocclusions for both moving and static scene elements (Section 8.3.1). The input video plus depth is used as the foreground layer. Object silhouettes are detected and cut in this layer, to allow for disocclusions. One background layer is computed by inpainting behind static occluders, and the other by background subtraction behind moving occluders. Since the depth maps provided by existing algorithms are not suitable for reprojection [330], we also introduce an algorithm that

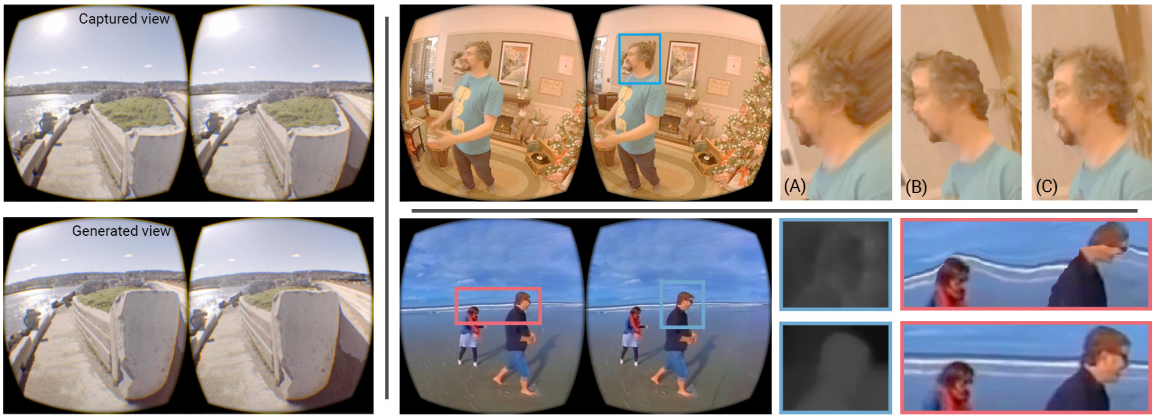


Figure 8.1: We add parallax for 360° videos, for viewing in virtual reality head-mounted displays (HMDs). This translates into a more compelling viewing experience, as our user studies confirm. *Left*: Captured point of view as shown in the HMD (top), and a novel view as the user moves their head (bottom); this novel view is generated with our method and was not captured by the camera. *Right, top row*: Straightforward approaches based on image-based rendering do not work well due to suboptimal quality of the depth information. Original view (left) captured with the GoPro Odyssey, and a close-up of novel views generated with three different methods (right): (A) naive reprojection of RGB information, (B) naive handling of disocclusions, and (C) our method, relying on a robust layered representation. *Right, bottom row*: We also propose a depth map improvement step to correct for errors that have a high impact on reprojection. Original view (left) from a YouTube video[24], and close-ups showing depth maps and a displaced view computed with them, for the original estimated depth map (top row), and for our improved depth map (bottom row).

preprocesses such depth maps to minimize visible artifacts (Section E.1). The algorithm cleans up occlusion boundaries and improves temporal coherence, leading to more visually plausible and appealing results.

Our approach provides a drop-in 6-DoF viewing experience that works for a wide class of existing 360° video available today, without any new capture or hardware requirements; we only assume that the video is captured by a static camera, which is common practice to minimize sickness during viewing. Although our method is not completely free of artifacts, we have performed three different user studies (Section 8.5), which confirm that it creates a more natural, compelling viewing experience, while also reducing the feeling of sickness or discomfort compared to traditional 3-DoF viewing. We provide source code for both our preprocessing algorithm and our real-time layered video viewer in our project website[25].

8.2 Related Work

IMAGE-BASED RENDERING Since the seminal works on image-based rendering [40, 79, 115, 223, 281], a number of IBR techniques have emerged that differ mainly either in the characteristics of the input data, or the type of scene representation used. Our work is related to this field, but our input differs substantially from what these works typically use.

A large group of works seek to allow free viewpoint navigation performing reprojection aided by some geometry proxy, often employing multiview stereo to obtain a 3D reconstruction [56, 79, 92]. To compensate for potential errors or sparsity in the 3D reconstruction, Goesele et al. [113] use *ambient point clouds* when rendering areas of the scene that are poorly reconstructed, while Chaurasia et al. [55] use variational image warping to compensate for sparse 3D information. Baricevic et al. [19] densify sparse depth information from monocular SLAM via an optimization approach in a live scene, i.e., without pre-capture or preprocessing. All these works are targeted at multiview

[25] http://webdiis.unizar.es/~aserrano/projects/VR-6dof

setups with a relatively large baseline, while our input is an RGBD video panorama. Still, we use some ideas from these works, like the need for soft visibility maps at depth boundaries to reduce artifacts [92, 253].

Some works do not rely on an explicit 3D reconstruction, but rather on dense image correspondences, for view interpolation [193, 201]. In contrast to them, we have the ability of generating novel unseen viewpoints, rather than interpolating between existing ones. Others augment well-known RGB inpainting methods [71] with depth-based terms [42, 43] for disocclusion filling in novel views; we partially rely on inpainting, but will show that simpler methods are better suited for our purposes.

Learning-based approaches have also been used to interpolate novel views, e.g. [103], or for light field acquisition [159], including light field video [333]. Closer to ours is the recent work by Zhou and colleagues [360] for wide-baseline stereo pair generation from narrow-baseline stereo input, using a deep neural network that infers a multiplane image representation of the scene; however, they do not handle dynamic scenes and their method is limited to stereo pairs.

The use of a layered representation is common in IBR approaches. Zitnick et al. [363] use a two-layer representation, in a system with a fixed, wide-baseline camera rig that allows them to obtain cleaner depth boundaries and mattes for areas near depth discontinuities while handling video. A more recent example is the work of Hedman et al. [135], who also use a two-layer representation for rendering. Follow-up work [134] improves reconstruction quality and computation times leveraging dual-lens cameras present in current high-end mobile devices. These two methods capture high-fidelity *static* 3D scenes, but they are not suitable for dynamic scenes nor video since the scene needs to be captured from many different points of view (some of the examples they provide in the dataset require dozens of images as input). In contrast, our approach allows automatic 6-DoF capture for dynamic scenes and videos, being agnostic to the capture hardware, including capture setups with a very narrow baseline. Another interesting approach is that of Philip and Drettakis [256], in which multi-view inpainting is performed on an intermediate representation formed by locally planar spaces shared between the input images; the work focuses on inpainting large unknown regions in large datasets with wider baselines than ours.

Finally, a series of works perform novel view synthesis to create content for autostereoscopic displays from stereo pairs, via image-domain warping [302]; using a steerable pyramid decomposition and filtering also used in motion magnification [82]; dealing with artifact detection and removal in synthesized stereo pairs [80]; or focusing on real time performance during the conversion [163]. In our case, the novel views we create can be from any viewpoint and do not follow a certain structure as they usually do in autostereoscopic displays.

6-DOF VIEWING A number of works have targeted creating a full 6-DoF experience in VR. Thatte et al. [318] propose a novel data representation to support motion parallax, *depth augmented stereo panoramas*, but it requires a specific capture setup, different from that of commonly available footage. Huang et al. [144] take as input a monoscopic 360° video with a moving camera, and, after doing a 3D reconstruction of the scene, are able to generate novel viewpoints by warping the initial 360° footage. They require, however, that the movement of the camera in the input video provides sufficient baseline for the 3D reconstruction. Visualization of real 360° scenes with head motion parallax is provided by the work of Luo et al. [199]; however, they require that a robotic camera arm captures the scene at specific positions, uniformly sampling a sphere in latitude and longitude, and they cannot handle dynamic scenes. Also addressing the problem of 6-DoF 360° content from light fields, Hinds et al. [140] proposed a benchmark for assessing light field compression techniques in the context of a view synthesis pipeline.

There has also been work on applying view-dependent texture mapping [77, 79] to speed up the rendering of the stereo pair in VR setups, assuming the input is a set of captured images suitable for a multiview stereo technique [269]. The recent work of Koniaris et al. [173] provides head motion parallax and view-dependent lighting effects in 360° content, but is targeted at synthetic content: the method takes as input a number of pre-rendered views of the scene of interest, as opposed to our real-world captured videos. Finally, Schroers et al. [279] presented a system that enables horizontal motion parallax from footage captured with a 16-camera rig: They reconstruct a dense light field panorama from a sparse number of input views. There are, however, three main differences with respect to ours: First, they require a calibrated camera rig for capture; second, they require a much larger amount of storage, since they need to store a separate 360° video stream for *every* discrete viewpoint; and third, they only provide horizontal parallax, which has been shown to yield a significantly worse viewing experience than full 6-DoF parallax [317].

In contrast to these works, we take as input a monoscopic 360° video of a scene captured from a static viewpoint, as given by typical 360° capture systems, plus a depth map obtained from the recorded footage. To our knowledge, only the work of Sayyad et al. [275] targets providing 6-DoF from a single panorama, but they do it via user intervention through an interactive modeling tool that allows the user to modify the geometry directly in VR; in contrast, we target an automatic approach that can also be suitable for video and dynamic scenes.

DEPTH MAP IMPROVEMENT TECHNIQUES Due to small baselines, specular or transmissive surfaces, or moving objects, among other causes, depth maps estimated from multiple images can be imprecise and contain artifacts. Image-guided filtering techniques provide depth maps that are well-aligned with the corresponding RGB edges; examples of local methods that can be used include bilateral filtering [20, 58], joint bilateral upsampling [174], guided filtering [131], or multilateral filtering [53]. However, RGB discontinuities may lead to depth artifacts that become clearly visible in 6-DoF viewing.

Global optimization methods can also be used for depth refinement. Levin et al.'s colorization based on optimization [188], which propagates scribbles according to the color affinity in the input image, has been applied to depth maps [135]. Image matting can also be used to propagate input scribbles [189]. These methods obtain good results for hole filling and sharp input edges, but this is not always the case in estimated depth maps from camera rigs or single-image methods.

We propose here a depth refinement technique that is inspired by these works but tailored for our end goal. For a more in-depth review of related techniques, we refer the reader to a recent survey on hole filling strategies for depth map completion [16].

There is another line of works that try to minimize contour artifacts when *reconstructing* depth from stereo or multiview content [28, 98, 295]. However, our input already includes a depth estimate along with the RGB frames, while building it from scratch applying these existing methods would likely fail due to the small baseline and minimal overlapping regions of current 360° cameras; for example, the work of Shan et al. [295] requires thousands of images. Further, this would limit our generality, since it would not work with monocular footage. Additionally, the works of Feris et al. [98], and Birchfield and Tomasi [28] are not intended for reprojection, and thus they do not consider geometry appearance from novel points of view.

8.3 Layered video representation

The input to our method is a 360° video, with RGB and depth values at each pixel. These can be initially provided by existing stitching algorithms [13], or estimated with a CNN [112]. Instead of using this depth information as is, which leads to visible artifacts when enabling 6-DoF, we first preprocess it and make it suitable for reprojection (see Section E.1).

The most basic reprojection algorithm is to convert the input video into a spherical triangle mesh, mapping each pixel to a vertex with the given RGB value, and 3D position as determined by the depth map, i.e., given a pixel at coordinates (θ, ϕ) on the equirectangular image, and depth d, map it to spherical coordinates (d, θ, ϕ). We use input videos with resolution 1024×2048, so this corresponds to a mesh of 1024×2048 vertices. Then, this mesh can be rendered at runtime, as the viewer moves their head. However, this *naive reprojection* approach produces very noticeable artifacts at disocclusion boundaries, as shown in Figure 8.1, close-up (A): As the viewer moves away from the center of the projection, the triangles of the mesh that correspond

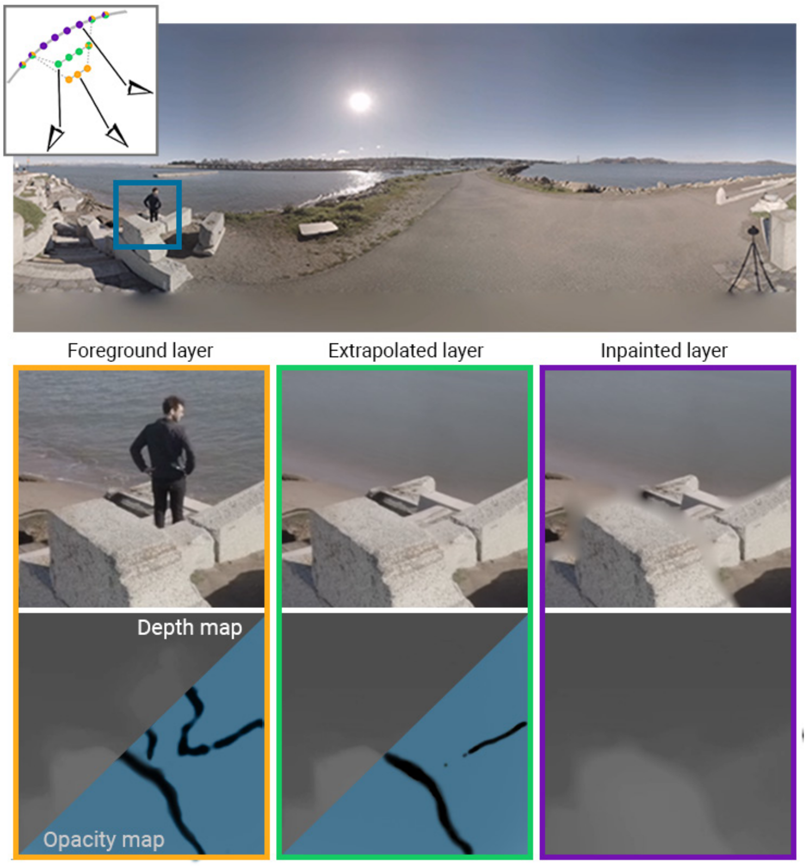


Figure 8.2: Example showing the layers in our scene representation. Top row: Sample RGB frame. The inset depicts a simple illustration of our three layers, showing which layer the user will see from different points of view (for a more detailed explanation see Figure 8.3, left). Middle row: For a close-up region, we show RGB of each layer (*foreground*, *extrapolated*, and *inpainted*). Bottom row: For the same close-up region, we show the associated depth map, and, for the two layers where it is present, the associated opacity map.

to disocclusion boundaries incorrectly connect foreground and background elements, rather than revealing disoccluded regions. To fix this, one might attempt to identify disocclusion boundaries, say, by depth differences, and then break the mesh at those boundaries. However, this naive handling of disocclusions can lead to jagged silhouette boundaries or missing information due to inaccuracies in the boundary estimation, in the depth map, or in the correspondence between depth and RGB edges, as shown in Figure 8.1, close-up (B).

We do employ a mesh-based approach, but making use of a layered representation that allows us to fix the aforementioned issues of missing information and jagged silhouettes in disocclusion boundaries. We describe this representation first (Section 8.3.1), then how it is computed (Section 8.3.2), and finally, how this representation is used to render the scene during real-time viewing (Section 8.3.3).

8.3.1 *Scene representation*

Our scene representation is designed to display the RGBD video with clean disoclussion boundaries at object silhouettes. In order to fill holes (missing information) in disocclusions, we use both background subtraction and inpainting where appropriate. We seek, as much as possible, to use the original video for rendering, e.g., rather than splitting objects in the video into separate layers. Finally, we wish to allow for real-time playback.

In order to achieve these goals, we introduce a layered representation. Our representation extends previous layered IBR approaches (e.g., [135, 363]); specifically, we extend the representation to three layers: a dynamic foreground layer, and two static background layers. The *foreground* layer is a mesh generated from the RGB video and its associated depth, together with an extra per-frame opacity map ($\hat{\alpha}^F$) used to control the opacity of the mesh at disocclusion boundaries. This layer is thus stored as an RGB video with an associated depth video and opacity video. There are two static background layers (which we term the *extrapolated layer* and the *inpainted layer*, respectively), used to fill disocclusion holes created when motion parallax takes place as the viewer moves their head. The extrapolated layer contains information for static background regions that are, at some time, occluded by moving objects (e.g., behind a moving person or car); hence, it can be computed by background subtraction. It also has a static opacity map ($\hat{\alpha}^E$) associated with it, again used to control transparency of this mesh when disocclusions occur. The layer is stored as an RGBA image and associated depth map. The inpainted layer contains information to fill-in disocclusions of areas corresponding to static background regions occluded by static objects. Those areas were thus never observed by the camera, and so must be computed by inpainting. Since this is the rearmost layer, it does not have an associated opacity map, and is stored as an RGB image and associated depth map. An example of these three layers is shown in Figure 8.2, and a visualization is given in Figure 8.3.

Given this representation, rendering the scene at a given time index entails converting the three RGBD images to three meshes as described above, and rendering each in the same 3D space. While the viewer's head remains at the center of projection, only the foreground layer will be seen. When the viewer's head moves from the camera center, the background layers will be visible at disocclusions. To enable this, the transparency of the foreground layer and extrapolated layer is computed making use of the opacity maps ($\hat{\alpha}^F$ and $\hat{\alpha}^E$, respectively). Section 8.3.3 describes rendering of the meshes during real-time playback.

Note that the static background layers cannot be merged because both layers may contain information in a given (θ, ϕ) direction. For example, suppose a person walks in front of a static car. From the camera center, three layers overlap: the moving person, which will be rendered in the *foreground layer*; the static car, which must be rendered in the *extrapolated layer*, since it is visible in other video frames; the scene behind the car, estimated by inpainting into the *inpainted layer*. This is illustrated in Figure 8.3 (left): depending on the user's point of view as they move, we will need to show either the *foreground layer* (viewpoint B), the *extrapolated layer* (viewpoint C), or the *inpainted layer* (viewpoint D).

8.3.2 *Layer computation*

We now describe how the layered representation is computed, including the three layers and the opacity maps.

8.3.2.1 *Foreground layer*

The foremost layer is dynamic and is the only one visible as long as the viewer's head does not move from the center of projection. It is directly generated from the original RGB video, together with the pre-processed depth (the pre-processing of the original depth is described in Section E.1).

The opacity map stores, for each vertex of the *foreground layer* mesh, a value $\hat{\alpha}^F \in [0..1]$ that will be used to control the opacity of the layer at runtime (Section 8.3.3). The idea is that, as the viewer's head moves from the center of projection, and disocclusions occur in certain areas, the *foreground layer* fades in those areas to allow visibility of the back layers. Thus, this map should store which vertices are likely to belong to disocclusion boundaries. Intuitively, disocclusion boundaries will be found at parts of the mesh whose normals are approximately perpendic-

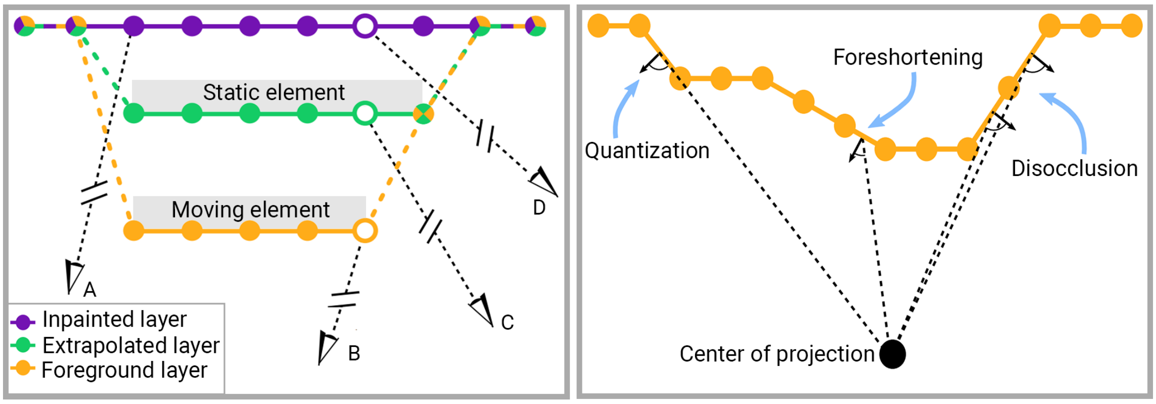


Figure 8.3: Left: Illustration explaining our three-layer representation. Each circle represents a different pixel, and its color indicates the layer it belongs to. Lines represent mesh connectivity (i.e., faces), and potential disocclusions are represented with dashed lines. The *foreground* layer shows a moving object and the *extrapolated* layer a static one. Some simple cases can be handled with only two layers (viewpoint A; black dotted lines represent a single ray for a given viewpoint). In others, however, three layers are required because depending on the viewpoint we the user should see the *foreground layer* (viewpoint B), the *extrapolated layer* (viewpoint C), or the *inpainted layer* (viewpoint D). Right: Illustration showing the relationship between face normals and potential disocclusions. The angle between these normals and the viewing direction from the center of projection (dashed lines indicate this direction for a number of faces) is used to compute the initial opacity map of a layer (O^F). We identify potential disocclusions as mesh triangles with angles close to 90°. Angles due to foreshortening will be narrower than angles belonging to disocclusions, since depth discontinuities are smoother. Quantization errors will produce sparse angles close to 90°.

ular to the *viewing direction from the center of projection*, since that is indicative of a sharp depth discontinuity and thus a potential disocclusion boundary (see Figure 8.3, right). Thus, by storing the angles between face normals and the view direction from the center of projection (in practice, the dot product between these two vectors) we would in principle have an initial foreground layer opacity map per frame, O^F. This initial opacity map, which is different for every frame of the input video, is shown in Figure 8.4(b) for a sample frame. Note that we leverage the GPU rendering pipeline, and compute the dot product in the fragment shader, i.e., with fragment normals (given the topology of our meshes, for each layer there is a single fragment per pixel of the input frame).

However, as also shown in Figure 8.3 (right), angles close to perpendicular can also be the consequence of foreshortening, or the result of quantization errors in the depth map. To minimize the impact of these situations, we apply to this initial opacity map O^F a closing morphological operation to remove subtle orientation changes due to depth artifacts, followed by a thresholding operation to remove smooth depth variations (most likely due to foreshortening or quantization effects), and then blur the result to provide a smooth transition at boundaries. Finally, a logistic function is applied in order to remove intermediate values that can cause ghosting, while still allowing for a smooth transition. Thus, the final opacity map of the *foreground layer*, $\hat{\alpha}^F$, is computed as:

$$\hat{\alpha}^F = S(G \circledast (\tau(O^F \bullet K))) \tag{8.1}$$

Operator • denotes the closing morphological operation, K is a disk kernel (radius of 2), τ denotes the thresholding operation (we use 0.8 as a threshold in all cases since O^F is in the range [0..1], 1 meaning perpendicular orientation), G is a Gaussian blur operator (kernel size of 7×7), and S is a logistic function (centered at $c = 0.5$, and with a slope $k = 8$), shown in Equation 8.2. Note that we apply all the operations in Equation 8.1 on the complement of O^F, and then take the complement of the result to yield $\hat{\alpha}^F$.

$$S(x) = \frac{1}{1 + e^{-k(x-c)}} \tag{8.2}$$

While we define our parameters heuristically, we use the same parameters for all input videos and all the different stitching methods and cameras we tested. Figures 8.4(c) and 8.4(d) show the result of applying these operations, while in Appendix E we include individual results for each of the operations described.

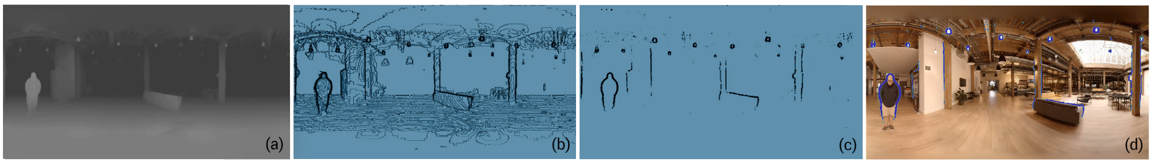


Figure 8.4: Example showing the computation of the opacity map. (a) Depth map of the *foreground layer* for a given frame of the video. (b) Initial opacity map O^F. (c) Final opacity map $\hat{\alpha}^F$. (d) Final opacity map shown in (c), overlaid on top of the corresponding RGB image. The orientation values O^F shown in (b) are too noisy to provide smooth opacity values, as they include information not only from potential disocclusions, but also from areas with foreshortening and quantization errors. We process these raw values (see text for details), and obtain a clean opacity map $\hat{\alpha}^F$, shown in (c), that smoothly matches potential disocclusion boundaries in the RGB image, shown in (d).

8.3.2.2 *Extrapolated layer*

This layer is static, and essentially contains a version of the scene with moving objects removed. We first compute its depth map, by taking, for each pixel, the n largest depth values of the *foreground* video frames, and compute their median to obtain a robust maximum depth (in all our tests, $n = 15$). Setting the depth to this robust maximum ensures that the observed background layer will remain behind the foreground layer, and thus invisible from the original camera viewpoint. To obtain the corresponding RGB values, we take, for each pixel, the RGB value associated to the selected depth value.

The computation of the opacity map for the *extrapolated layer*, $\hat{\alpha}^E$, is analogous to that of the *foreground layer opacity map* previously described. The only difference is that in this case we use the mesh of the *extrapolated layer* when computing the angles between the mesh normals and the view direction from the center of projection to yield the initial opacity map O^E. As such, this opacity map does not need to be computed per frame of the input video, but just once.

8.3.2.3 *Inpainted layer*

This is the backmost layer; it is also static, and contains inpainted regions in areas behind static objects that can become visible due to disocclusions. We obtain this layer by identifying which regions can become disoccluded, and inpainting them. The identification is done based on the opacity maps, whose computation is described below. A theoretical derivation to compute the maximum parallax that the layers may be subject to, and thus the largest area requiring inpainting, can be found in Appendix E. For inpainting we use a well-established PDE-based inpainting method [25] that smoothly interpolates the unknown values. We favor smoothing approaches over patch-based ones, since, upon failure, the latter are prone to produce very prominent local artifacts that are very distracting during playback, making our smooth approximation preferable for our particular application; a comparison of our result to a patch-based approach [186] is shown in Figure 8.5.

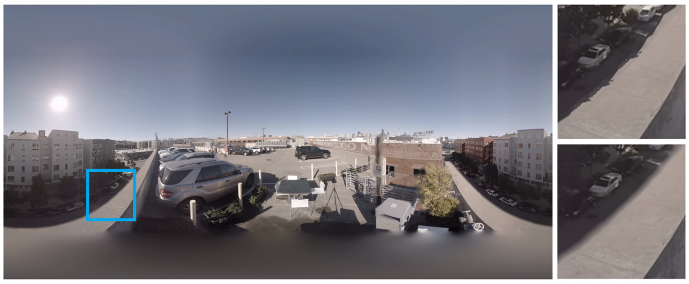


Figure 8.5: Left: RGB panorama. Right: Corresponding *inpainted layer* obtained using a patch-based approach [186] (top), and our result using a spring metaphor-based method [25] (bottom). The patch-based approach produces local artifacts that are very prominent when visualized on an HMD. While the spring-metaphor inpainting yields a blurred approximation in the area to inpaint, for our application this smoothness is preferable.

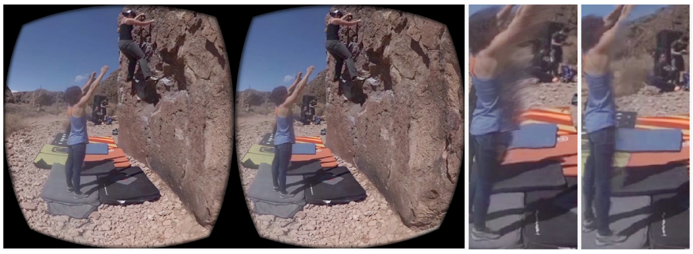


Figure 8.6: Comparison of two different methods to modulate the opacity values ˆ α^Fas a function of distance to the center of projection to obtain α^F. Left: Original view as seen in the HMD. Right: Close-up of a displaced view with two different modulations: linear interpolation (left), and blending based on a logistic function (right). Aggravated ghosting results from using a linear interpolation as a function of distance.

8.3.3 *Real-time playback*

During playback, the three layers are rendered as meshes. When the viewer is at the center of projection, only the foreground layer should be visible. As they move away from the center, we need to show information from the other layers as well, depending on the disocclusions that take place, as given by the precomputed opacity maps.

To model this behavior, at runtime, we modulate the opacity of each layer given by the opacity maps with a sigmoid function of δ, the current distance between the viewer's head position and the center of projection. Specifically, the run-time foreground layer opacity α^F is given by:

$$\alpha^F = S(\delta)\hat{\alpha}^F + 1 - S(\delta), \tag{8.3}$$

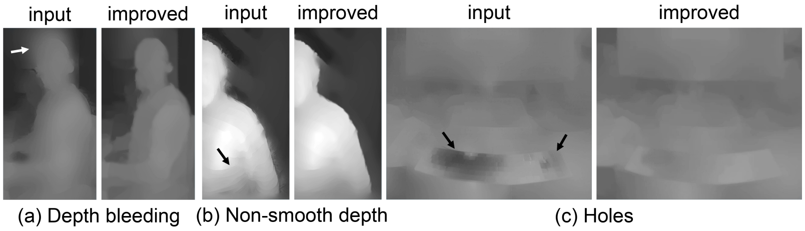
where $S(\cdot)$ is the function shown in Equation 8.2, with $k = 30$ and $c = 0.15$ meters for all scenes tested. The run-time extrapolated layer opacity α^E is obtained in an analogous manner, using its corresponding precomputed opacity map $\hat{\alpha}^E$. Note that, in this way, for $\delta \to 0$ the opacity of the layers tends to one (i.e., only the foreground layer is visible). As δ increases, the opacity of the layers is given by the opacity values stored in the pre-computed opacity maps. We choose blending with a logistic function (as opposed to using, e.g., a linear interpolation) since overly smooth transitions tend to result in aggravated ghosting, as shown in Figure 8.6.

8.4 Depth Improvement for Motion Parallax

This section describes the depth map preprocessing that we perform, in order to improve scene appearance at boundaries.

Our method relies on depth information from the scene. For scenes where a depth map is provided, such as output by a 360° stitching algorithm [13], we can use the provided depth map. Otherwise, we use an off-the-shelf neural network depth estimation algorithm [112].

However, the depth maps provided by existing algorithms are not optimized for reprojection, as also noted by Waechter and colleagues [330]. For example, both stitching accuracy and benchmark scores are relatively insensitive to slight variations in the pixels around object silhouettes, since they are a tiny subset of any depth map. But small errors in the silhouette depths leads to extremely objectionable ragged boundaries when reprojecting to new viewpoints. On the other hand, small depth errors in object interiors are hardly noticeable. Other artifacts include depth bleeding across silhouettes, strong discontinuities in what should be continuous surfaces, and temporal inconsistencies (Figure 8.7). This is a general problem that applies to all current approaches to depth map estimation.



(a) Depth bleeding (b) Non-smooth depth (c) Holes

Figure 8.7: Examples of artifacts in the input depth videos that hamper the viewing experience, and our resulting improved depth. (a) Bleeding artifacts around object boundaries. (b) Piece-wise discontinuities in what should be smooth gradients. (c) Strong discontinuities in what should be a continuous surface.

Our goal in this section is then not to provide an algorithm that improves the accuracy of the depth maps in general. Instead, we focus on minimizing the three main sources of artifacts, thus making the scene look much more visually plausible and appealing.

We pose improving the input depth map as an optimization problem. The objective function has a data term (E_{data}), and constraints for edge preservation (E_{e}), spatial smoothness (E_{sm}), and temporal consistency (E_{t}):

$$\underset{d}{\operatorname{argmin}} \; \lambda_{\text{data}} E_{\text{data}} + \lambda_{\text{e}} E_{\text{e}} + \lambda_{\text{sm}} E_{\text{sm}} + \lambda_{\text{t}} E_{\text{t}}, \tag{8.4}$$

where d is the depth map of a frame of the video, $d(\theta, \phi)$; in the following, to simplify notation, we will use an index i to denote each (θ, ϕ) pair, that is, each pixel in each equirectangular-format frame. The optimization is solved per frame. We chose the λ values empirically (an evaluation can be found in Appendix E), for our experiments $\lambda_{\text{data}} = 0.1$, $\lambda_{\text{e}} = 1$, $\lambda_{\text{sm}} = 0.5$, and $\lambda_{\text{t}} = 0.1$. Note that we pad both RGB and depth maps with the corresponding wrap-around values of the equirectangular projection.

The **data term** ensures fidelity to the input depth values:

$$E_{\text{data}}(i) = \sum_{\text{i}} w_{\text{d}}(i) \left(d(i) - \hat{d}(i) \right)^2, \tag{8.5}$$

where $\hat{d}$ denotes the input depth. The per-pixel weight $w_{\text{d}}(i)$ models the reliability of the input data. Its aim is to reduce data fidelity along edges, since error in the input depth is more prominent in those regions. It is based on the local variance of the depth σ^2 of each pixel i (the higher its variance, the less reliable). Specifically: $w_{\text{d}}(i) = e^{-\gamma \cdot \sigma^2}$, where γ is set to 10^5 for all cases, and the window size to compute σ^2 is 7×7.

To enforce clean edges, we add an **edge guidance term** that penalizes propagation across edges by using the edge weight $w_e(i, j)$, inspired by the colorization method of Levin et al. [188]:

$$E_e(i) = \sum_i \left(d(i) - \sum_{j \in \mathcal{N}(i)} w_e(i, j)\, d(j) \right)^2 , \tag{8.6}$$

where j denotes pixels in a neighborhood of pixel i. Unlike Levin et al.'s method, we use $w_e(i, j) = \tau(e(i) - e(j))$, where e represents an edge vote map and τ is Tukey's biweight [29] (comparisons with different functions for τ can be found in Appendix E). To compute the edge vote map e, we use a multiscale edge detector [84], taking into account edge information from both the RGB image (e_{rgb}) and its corresponding input depth map (e_d), so that $e = e_{\text{rgb}} + e_d$. As a result, edges corresponding to a depth difference will have higher edge vote map values, and thus lower weights (w_e).

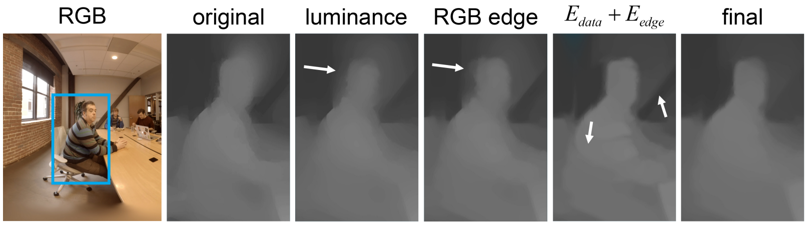


Figure 8.8: Depth improvement results for different variations of our optimization. The first and second columns show the input RGB and depth images, respectively. The third and fourth columns show the results using luminance or an edge map computed from RGB as guidance. In the fifth column we show the result when removing the smoothness term (Equation (8.7)) from our optimization. As the black arrows indicate, clear artifacts remain in all three results. The last column shows the result of our optimization.

The **smoothness term** acts over local neighborhoods:

$$E_{sm}(i) = \sum_i \sum_{j \in \mathcal{N}(i)} w_{sm}(i) \left(d(i) - d(j) \right)^2 , \tag{8.7}$$

where w_{sm} is the smoothness weight, obtained in the same way as the data weight w_d, by computing the local variance σ^2 of the input depth map: $w_{sm}(i) = e^{-\beta \cdot \sigma^2}$. In this case σ^2 is computed within a 3×3 neighborhood, and we set $\beta = 10^3$ for all cases. The weight is lower the higher the variance within a local neighborhood, so that smoothness is only imposed in regions where there are no abrupt changes in depth. Figure 8.8 shows the influence of each term in our improved depth, along with the result using luminance or RGB values.

Finally, **the temporal consistency term** is defined as:

$$E_t(i) = \sum_i w_t(i) \left(d(i) - \psi_{\text{prev} \to \text{cur}} \left(d_{\text{prev}}(i) \right) \right)^2 , \tag{8.8}$$

where $\psi_{\text{prev} \to \text{cur}}(\cdot)$ is a warping operator between two frames, implemented as the variational robust optical flow method [194]. The weight $w_t(i)$ is computed as: $w_t(i) = \max \left(\varepsilon, \sqrt{u(i)^2 + v(i)^2} \right)$, where ε is set to 10^{-4}, and $u(i)$ and $v(i)$ correspond to horizontal and vertical flows at pixel i. This weight is higher if there is larger motion, since temporal artifacts would be more noticeable.

Since all the terms are l_2 norms, we solve Equation (8.4) with the conjugate gradient method. For time efficiency, we first downscale the input RGB and depth to 80% of the original size in each dimension, then we upscale the refined depth to the original resolution, followed by a fast bilateral filtering [58]. Figure 8.9 compares our depth refinement step with other common existing approaches; our method provides cleaner depth maps, which are a key factor when adding parallax cues.

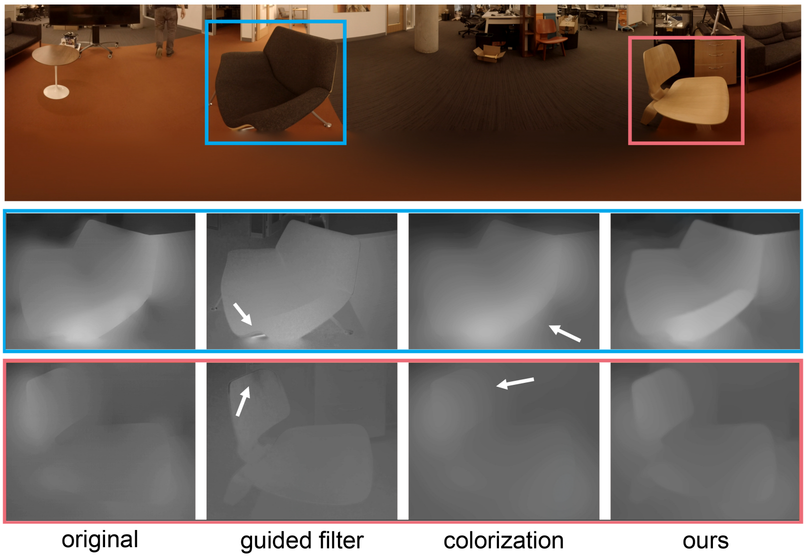


Figure 8.9: Comparison of our depth improvement against guided filter [131], and colorization [188]. Since the former is guided by an RGB image, it leads to artifacts similar to those shown in Figure 8.8, whereas the latter tends to over-smooth the result.

8.5 Evaluation

Recents studies suggest that 6-DoF provides an overall better viewing experience (e.g., [317]). To validate whether the results achieved with our method also provide an advantage over conventional 3-DoF 360° video viewing, we perform three different user studies using 360° stereo videos with and without motion parallax. Note that for both conditions (3-DoF and 6-DoF) we display stereo views. Specifically, we want to answer two key questions: (a) Does our added motion parallax provide a more compelling viewing experience?, and (b) does it reduce sickness? For the first question on *preference*, we designed two experiments, carried out first and last. In between, we performed the *sickness* experiment. Since the naive handling of disocclusions (Figure 8.1, close-up (B)) yields very noticeable artifacts, we chose not to include it as a condition in our studies.

All 360° videos were shown on an Oculus Rift connected to a PC equipped with an Nvidia Titan, running at 90fps. Similar to VR applications that use a limited tracking volume, we constrain the maximum displacement from the center of projection using visual cues. To first find what makes a reasonable range of casual, accidental motion, we carried out an initial study in which we registered head movements when watching our 6-DoF 360° content. We define accidental motion as involuntary head translations that will occur during rotational movements in a natural exploration of the scene. Figure 8.10 (left) shows the resulting histogram, measuring the Euclidean distance to the center of projection 90 times per second; we observe that most movement is clearly constrained to a certain range. Based on this histogram, we set two thresholds at 20 and 35 cm. When the first threshold is surpassed, blue latitude circles are overlaid to the video content, and the image progressively fades to gray (see Figure 8.10, right); once the second threshold is surpassed, the image fades to black. For fairness, when comparing between our method and conventional 3-DoF viewing, we added the visual cues for constraining the displacement to both. Participants were informed about this visualization previously to all experiments.

PARTICIPANTS A total of 24 participants (8 female, 16 male), ages 18 to 20, took part in the experiment. The same subjects participated in the three studies, carried out on three separate days to avoid fatigue and accumulation effects. They voluntarily signed up for our experiments, were naive with respect to their purpose or our technique, and were paid 25$ upon completion of the experiment on the third day. They were first asked to answer a brief questionnaire about their previous use of HMDs and VR content: 14 subjects had no previous experience with HMDs or VR content, 8 subjects had used an HMD less than 3 times before, and 2 subjects had used an HMD up to 10 times before.

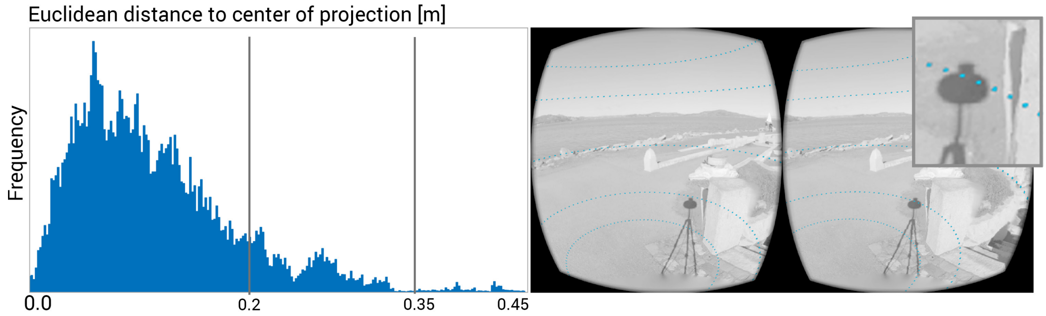


Figure 8.10: Left: Histogram of the Euclidean distance from the head position to the center of projection during our pilot experiment. We observe that most movement is limited to a certain range. The two gray lines indicate the two thresholds that control our visualization: at 0.2 m, blue latitude circles appear and the scene progressively fades to gray (shown on the right); at 0.35 m the scene fades to black.

8.5.1 *Experiment #1: Preference (part I)*

In the first experiment we evaluate whether our method provides a more compelling viewing experience.

STIMULI The stimuli consisted of seven 360° videos, covering a variety of scene layouts, content, and motion. For each video, there were two versions: the original one, and adding motion parallax using our method. The videos were captured with a GoPro Odyssey and a Yi Halo cameras, and stitched using the algorithms provided by Google Jump Manager. We show some representative frames in Figure 8.11, please refer to Appendix E for more details. To keep the subjects engaged, we limited the videos to 30 seconds each and, following common practice [44, 158], participants were informed that they would be asked a few questions about the scenes after watching them.

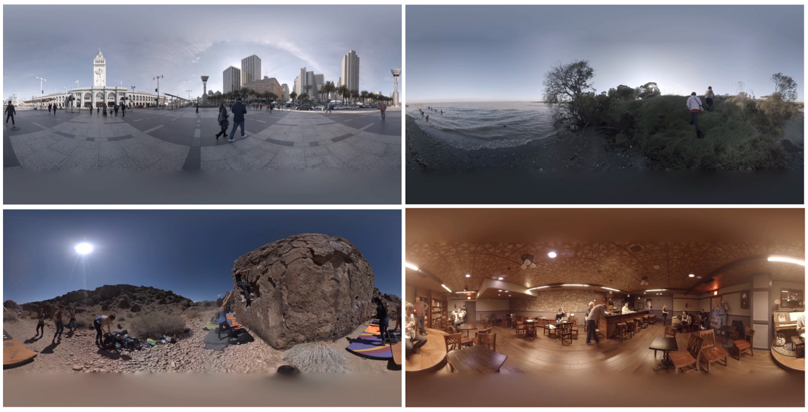


Figure 8.11: Example frames of the videos used for our user study.

PROCEDURE Each participant watched the seven videos twice, once without (conventional 3-DoF viewing) and once with motion parallax added with our method. The order was randomized for each video, separated by a one-second black screen. After seeing each pair of videos, they answered a questionnaire where they had to choose one method or the other in terms of realism, comfort, immersion, presence of visual artifacts, and global preference. There was also a space for comments, allowing the participants to explain the reasons for their answer. The whole questionnaire can be found in Appendix E. At the end of the experiment, a debriefing was carried out, but participants were not told the goal nor any information about the techniques tested in the experiment, since they had to go through two more sessions in the next days. The whole experiment took less than 30 minutes per subject.

RESULTS We show in Figure 8.12 (left) the results for the global preference question. The videos with added parallax using our method were preferred in six out of the seven cases. Additionally, seven users commented about the movement and the depth being "more realistic" with our method.

8.5.2 *Experiment #2: Sickness*

One of the main reasons motivating our technique is the hypothesis that, when watching 360° video, the absence of motion parallax may induce a feeling of sickness or discomfort, due to the mismatch between different sensory inputs (visual and vestibular). In our second experiment, we set out to measure if this is the case, and if it is less prominent when adding motion parallax with our technique.

STIMULI The stimuli consisted of a set of 12 videos, distinct from those in the previous experiement, but again covering a wide variety of scene layouts and movements. The videos were captured with a GoPro Odyssey and a Yi Halo cameras, and stitched using the algorithms provided by Google Jump Manager. The videos lasted between 30 seconds and one minute each, for a total duration of 8 minutes and 30 seconds.

PROCEDURE We created two blocks with the set of 360° videos; the first contains the 12 original videos without motion parallax, while the second features motion parallax with our method. There was only a brief pause of $200ms$ between videos, because we wanted to analyze the exposure to a continuous 360° viewing experience. This experiment consisted of two sessions on two different days. On the first session, they watched one block and answered a questionnaire containing (a) the VRSQ questionnaire (Virtual Reality Sickness Questionnaire) [166], and (b) two yes/no questions asking whether they had experienced, at any time during viewing, sickness, dizziness, and/or vertigo, and whether they had experienced discomfort; please refer to Appendix E for the full questionnaire and details. They were also instructed to describe the video in which they felt such symptoms. On the second, they did the same for the other block. The order of the blocks was randomized between participants. The participants were allowed to stop if they needed to, but none requested to do so, and a debriefing followed the sessions. The whole experiment took 15 minutes per session.

RESULTS The results of this experiment indicate that our method does indeed help in reducing sickness by enabling motion parallax: while 17 out of the 24 participants reported symptoms of sickness, dizziness, and/or vertigo while watching the videos without motion parallax, only 5 experienced these symptoms with our method. Moreover, none of the subjects reported visual discomfort with our method, while 4 users experienced discomfort with conventional 3-DoF viewing. The results for the VRSQ questionnaire were inconclusive, possibly due to the short duration of the viewing session and the lack of a physical task (the authors of the VRSQ questionnaire [166] report sessions of 90 minutes, and different target selection tasks throughout the session).

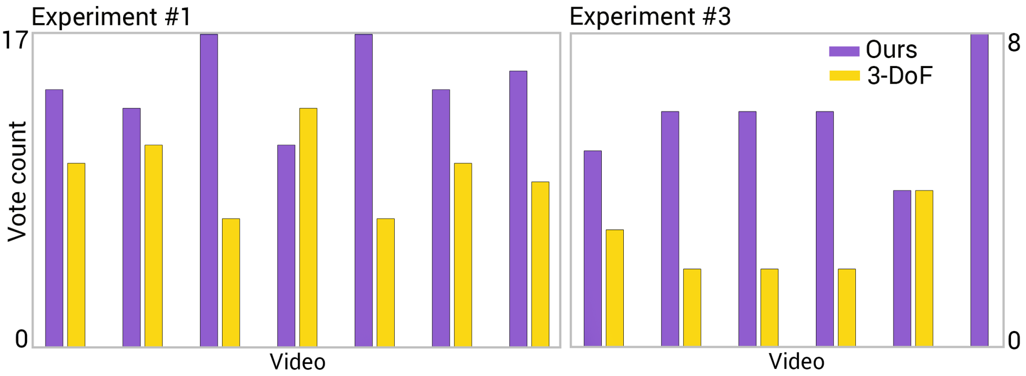


Figure 8.12: Vote counts for the global preference question in Experiment #1 (left) and Experiment #3 (right), for our 6-DoF method (purple), and the conventional 3-DoF (orange). The x-axes show the different videos tested. In Experiment #1, the videos with added parallax were preferred in six out of the seven cases, with a strong preference in two cases. In Experiment #3, our method was strongly preferred for five out of the six videos.

8.5.3 *Experiment #3: Preference (part II)*

In the last experiment, we repeated the procedure of Experiment #1 but this time disclosing in advance the difference between the two versions of each video (motion parallax). The goal was to test if, once the presence of motion parallax is explicitly known, users would find the viewing experience more or less enjoyable than before, and whether this altered their viewing behavior. We first played a video on a conventional desktop display showing a recorded HMD viewing session with and without motion parallax, verbally explaining the difference. Participants then put on the HMD to view the same scene, asking them to experiment moving their head. This process went on until we were sure that the participants understood the differences between the two viewing modes. This scene was only used for the explanation and was not included in the test set. This third experiment was carried out the last day, upon completion of the two previous experiments, to ensure that during the previous sessions they were unaware of the differences between the methods. Note that participants were still not aware that we were testing a new method, only two different viewing options.

STIMULI Each subject was presented with a random subset of two videos from Experiment #1 (not including the one used during the explanations). Each video was thus viewed and evaluated eight times.

PROCEDURE After the initial explanations, the procedure was the same as Experiment #1. Since each subject only watched four videos from two scenes, the total duration of this experiment was 10 minutes.

RESULTS As Figure 8.12 (right) shows, our method was strongly preferred for five out of the six videos, with no clear preference for the sixth. Furthermore, several participants verbally expressed and confirmed their preference, commenting that our method provided “a more realistic 3D experience”, that it "greatly helps the feeling of immersion”, and that “the movement is closer to that of the real world”.

8.5.4 *Analysis of viewing behavior*

To gain additional insights about the possible influence in viewing behavior of enabling motion parallax, we have analyzed the differences in head movement. For each trial in Experiments #1 and #3, we aggregated the distance from the head to the center of projection across the total viewing time, and performed a dependent t-test to compare the means of the distributions with and without motion parallax. We have found a statistically significant ($t(209) = 2.395, p = 0.018$) difference, indicating that, on average, users displace their head 4.3 cm more every second when 6-DoF are enabled. A possible explanation is that motion parallax allows for a more natural viewing of the scene, and thus fosters exploration. This may be also indirectly influenced by the significant reduction in sickness reported in Experiment #2.

We further analyzed head movement for comparing head movement differences between Experiment #1 and Experiment #3 (i.e., before/after explaining the differences between the methods). We followed the same procedure as in the previous analyses, and we found a statistically significant ($t(83) = 3.484, p = 0.001$) difference in the means. On average, users displaced their head from the center 13.46 cm every second more after knowing the differences between the methods. This would offer an explanation for the difference in preference votes with respect to Experiment #1.

Last, we analyzed separately head movement in Experiment #2, since the nature of this experiment was different from the other two. We followed the same procedure, and found that the results are consistent: there was a statistically significant ($t(275) = 3.352, p = 0.001$) difference, with users moving their head from the center 4.05 cm more with our method than with conventional 3-DoF viewing.

8.6 Results

We have tested our method in a variety of scenarios providing different kinds of input, including the challenging case of capture systems that do not yield depth maps. This is particularly important, since monocular 360° cameras (e.g., the Ricoh Theta) are more affordable and widespread than more sophisticated camera rigs. In the absence of an input depth map, we first use a Convolutional Neural Network (CNN) to estimate per-frame depth maps from monocular RGB [112]; however, the depth map from the CNN is not of sufficient quality for our purposes, and lacks temporal consistency. Our depth improvement stage significantly increases the quality of the final depth, including temporal coherence, thus enabling motion parallax even from such limited input.

We run the preprocessing of our input RGBD videos in a standard PC equiped with an Intel $i7-3770$ processor (up to 3.90 GHz). The processing time (on average) for our videos (resolution of 2048×1024 pixels) in a single core is: 7.71 seconds per frame for the *depth improvement* step, 762 miliseconds per frame for extracting and processing the *opacity maps*, 319 miliseconds per frame for computing the *extrapolated layer*, and 52.66 seconds (total) for computing the *inpainted layer*. The storage overhead adds to the original RGBD video, two static RGBD images (*extrapolated layer* and *inpainted layer*), a 360° video stream with the *opacity map* corresponding to the *foreground layer*, and an additional image corresponding to the *opacity map* of the *extrapolated layer*. After this processing step, our system runs in real time, providing the 90fps recommended for a satisfactory VR experience.

To test how well our method performs given different input depth maps from different capture systems, this section includes results from the GoPro Odyssey and Yi Halo cameras, both stitched with Google Jump Manager (Figures 8.1 and 8.15), from Facebook x24 (Figure 8.14), and from monocular videos from different sources with estimated depth (Figures 8.1 and 8.13). As discussed in Section E.1, we remind the reader that the depth maps produced by these methods have not been designed to help generate motion parallax effects; as such, they lead to obvious geometric distortions when generating novel views, due mainly to their ragged edges, the presence of holes, and temporal inconsistencies. As an example, Figure 8.1 uses Google Jump Manager algorithm for stitching and depth generation: however, distortions are still obvious in the novel views (rightmost scene, insets A and B) before applying our method (inset C). Figure 8.14 shows a result from Facebook x24: our depth refinement and smooth disocclusion handling method leads to the satisfactory computation of novel views.

Three more results are shown in Figure 8.15 (more results available in Appendix E), depicting a variety of scenes. For each result, we show a view from the center of projection, and a displaced view leading to disocclusions. In each of the scenes we highlight regions illustrating the added parallax. Last, we further illustrate the use of layers in Figure 8.16, where, given a displaced view, we color-code the pixels rendered from each of the three layers.

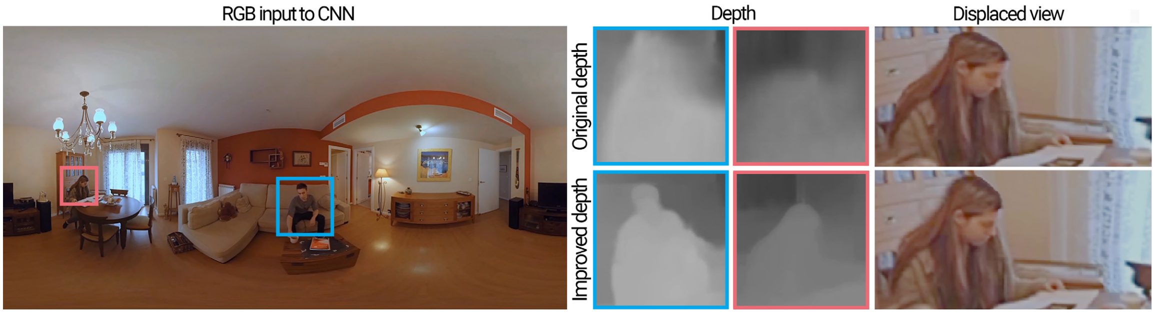


Figure 8.13: Result using monocular video without depth as input. We show a representative frame (left), details of the initial estimated depth [112] and our improved depth (middle), as well as the corresponding result when generating a novel view (right). Our method yields minimal distortions even in the presence of such suboptimal input. The scene was taken from a previously existing short clip (dataset from [292]); we use only a 360° RGB monocular view as input and drop the remaining data.

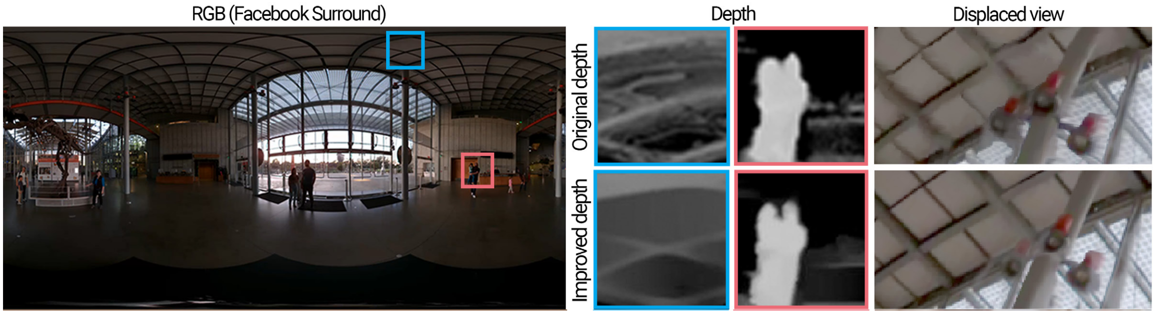


Figure 8.14: Left: Representative frame of a video captured by the Facebook x24 camera. Center: Comparison of the original depth provided by Facebook, and our improved depth for the two highlighted regions of the scene. Right: Our improved depth yields better reconstructions of novel views, without distracting artifacts.

8.7 Discussion and Conclusions

We have presented here a technique to enable head motion parallax in 360° video, thus enabling 6-DoF viewing of real-world capture footage. We have designed our method to be independent of a specific hardware, camera setup, or recorded baseline, showing examples from different common 360° capture systems, including depth estimated from scratch by a neural network.

We assume that each RGB (and depth) frame is a monocular panorama in equirectangular projection, but other projections (such as cube map) are also possible.

Our system requires only RGBD 360° video as input, and is rather robust to inaccuracies in the depth. Thus, it can deal with different data sources, from 360° video plus depth stitched and computed from individual videos from a camera rig, to 360° monocular video with depth

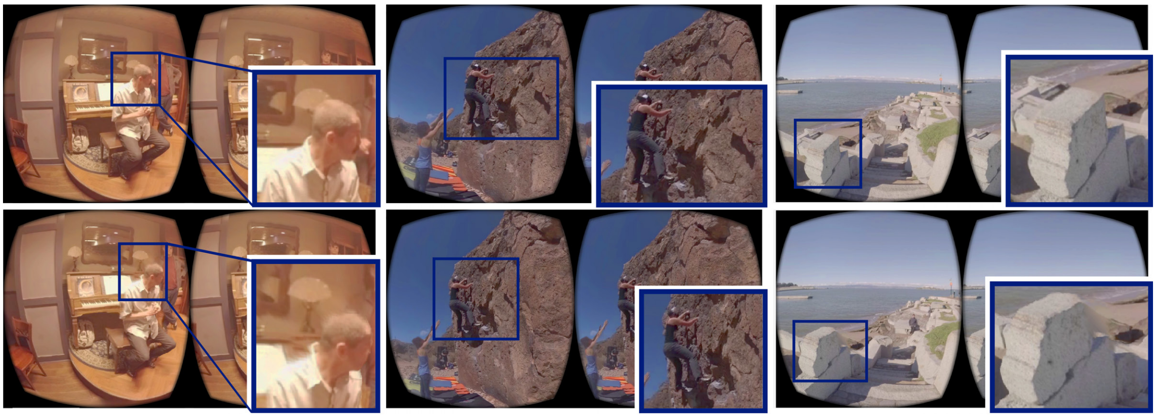


Figure 8.15: Three examples of novel views generated with our method. For each example we show the original view (top), and the corresponding displaced view (bottom). We also include close-ups of regions where the added parallax is clearly visible.

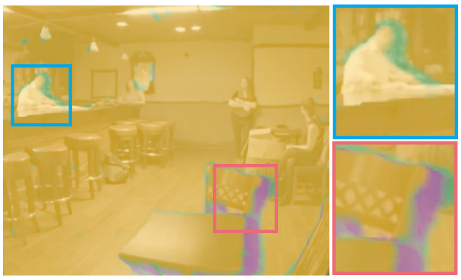


Figure 8.16: Color-coded visualization depicting the use of our three-layer representation in a given frame for a displaced view (orange: *foreground layer*; green: *extrapolated layer*; purple: *inpainted layer*).

computed with a CNN-based depth estimation algorithm. While having access to a multiview setup may not be commonplace, a number of 360° cameras do provide stereo output, which can yield a reasonable depth. Still some common cameras do not provide stereo (Ricoh Theta, Samsung Gear 360, or Nikon Keymission 360, among others), due, e.g., to the limited overlap between the camera views. Our method is designed to be agnostic to the hardware employed during capture, in order not to diminish generality.

Our user studies confirm that our method provides a more compelling viewing experience, while reducing discomfort and sickness. Interestingly, the additional degrees of freedom enabled by our method also influence viewing behavior: On average, users displace more their head from the center of projection when viewing content with 6-DoF.

LIMITATIONS AND FUTURE WORK The assumption of a static camera for the input video is reasonable in our scenario, since a considerable amount of 360° content is shot with static cameras. HMD and 360° camera manufacturers typically recommend static cameras [95, 241, 306, 316], and static cameras are widely preferred to moving cameras for most types of 360° videos to reduce potential sickness.

This assumption is mainly required due to the way we compute the extrapolated layer; further, if the camera moved we would need the extrapolated and inpainted layers to be dynamic (i.e., videos instead of images), requiring more storage and bandwidth during playback. Aside from this, our representation could be extended to handle a moving camera, and we leave this to future work.

Our layered representation features three layers. In theory, more than three layers could be needed depending on scene complexity, and this would incur additional storage and processing requirements. The number of moving and/or static objects overlapping in time and space, and in general, the depth complexity of the scene, would need to be assessed together with the increase in algorithmic complexity, in order to choose the optimal number of layers for each scene. In practice, however, we find our solution to be enough, and a number of reasons support our choice: First, three layers represent the types of motion present in many 360° videos: a few moving foreground actors or objects, as well as static objects; second, the amount of storage and bandwidth would increase with the number of layers, eventually hindering real time playback; third, for scenes with greater layer complexity, determining the number and content of layers would be very challenging from 360° video alone, and not necessary for typical videos given the amount of head motion we target, and that has been shown as usual in these setups [317].

Our computation of the extrapolated layer has an implicit limitation for cases with strong lighting variations (e.g., moving shadows). In those cases, depth remains constant but RGB can vary significantly, so we can have some "noise" in the extrapolated layer. In practice, however, the small extent and varying nature of disocclusions result in this effect being negligible. Additionally, there might be some cases in which large scene objects are close to the camera and remain stationary. In such cases, one would need to resort to the *inpainted layer*, and more complex inpainting algorithms may be needed.

Our method relies on the quality of the input depthmap, especially near disocclusion boundaries. We lessen this dependency with our depth improvement step, however, our method does still introduce some artifacts at disocclusion boundaries, which is to be expected given the extremely limited nature of our input. Our studies show that users prefer 6-DoF viewing to 3-DoF viewing, even with these artifacts; getting rid completely of such artifacts remains an open challenge. It is possible that combining ideas from our work and the works by Hedman et al. [134, 135] could lead to higher-quality 6-DoF capture.

These errors increase as the viewer moves farther away from the center of projection. In the future, we would like to explore options to minimize this, for instance by creating a non-linear mapping between the head and the camera movement that prevents the viewer from moving too far. Alternatively, existing techniques for controling user attention in VR [74, 117] could be helpful in this context. Last, as a consequence of the omnidirectional stereo (ODS) format, depth information near the poles is not accurate [267], and thus the performance of our method worsens near those regions. However, these regions are rarely observed due to a horizon bias [298].

Many studies exist assessing presence, immersion, or discomfort when exploring virtual reality (synthetic) content. In contrast, existing studies on viewer experience watching 360° footage on HMDs are preliminary and much remains to be explored about how we should display real content on immersive HMDs. We hope that our work provides a solid background for subsequent studies in this area.

Part V

Conclusiones and future work

9. Conclusions and future work

In this thesis we have presented a series of contributions on three areas of visual computing: computational imaging, material appearance, and user behavior in virtual reality. In the first part we have focused on efficient acquisition of the plenoptic function; this has led to contributions in high-speed video and high dynamic range image capture. In the second part, we have attempted to address the long-standing problem of perception and editing of real world materials, proposing a perceptually-based control space for captured data, and later focusing on using this space for the particular application of gamut mapping. Finally, in the third part, we have taken the first steps towards understanding user behavior and attentional mechanisms in virtual reality, starting with simple static stimuli, and moving later to complex dynamic environments in a cinematographic context. In the following we summarize the conclusions and future work for each of the three parts.

COMPUTATIONAL IMAGING In this first part we have presented two main lines of work. In the first one (Chapter 2) we have presented a framework for reconstruction of HDR images using convolutional sparse coding. From a single, optically-coded image, we reconstruct dynamic range using a trained convolutional filter bank. Our approach follows a current trend in computational photography, leveraging the joint design of optical elements and processing algorithms. Once trained, the obtained filter bank can be used to reconstruct a wide variety of HDR images greatly differing from the training set. Since our reconstruction is based on a convolutional approach, it does not rely on the linear combination of patches common in sparse reconstruction methods; this greatly reduces reconstruction artifacts, in particular in high-contrast sharp edges present in HDR images. We are not limited to a restricted number of captured exposures, nor do we face the implicit trade-off between captured dynamic range and interpolation quality that other methods based on spatially-varying exposures face. As an additional advantage, our framework naturally extends to HDR video capture. As future work, the development of patents like Sony's per-pixel, double-exposure method will progressively introduce varying exposure and optically modulated systems, thus allowing for increased capabilities of commercial cameras. Our optimization could incorporate explicit modeling of image noise to perform denoising in particularly noisy images.

In the second one (Chapter 3) we have focused on the particular case of high-speed video acquisition, and the intrinsic trade-off between temporal and spatial resolution imposed by bandwidth limitations. We have presented two sparse coding approaches, where we code the temporal information by sampling different time instants at every pixel. We have analyzed the key parameters in a patch-based sparse coding approach, allowing us to offer insights that lead to better quality in the reconstructed videos. We then have introduced a novel convolutional sparse coding framework. The convolutional nature of the filter banks used in the reconstruction allowed for a more flexible and efficient approach, compared with its patch-based counterpart. We bypass the need to capture a database of high-speed videos and train a dictionary, while reconstruction times improve significantly. Many exciting venues for future research lie ahead. For instance, our strategy to impose an additional constraint in the temporal dimension is motivated by the fact that, due to the size of the convolutional matrices required to train the dictionary, it is not feasible to deal with (x-y-t) blocks directly. This is currently the main limitation of our approach, and it would be interesting to investigate other strategies in follow up work.

Finally, an exciting avenue of future work lies at the convergence between acquisition and display technologies, for the full plenoptic function and taking perceptual considerations into account. Computational imaging aims at enhancing imaging technology by means of the co-design of optical elements and algorithms; capturing and displaying the full, high-dimensional plenoptic function is an open, challenging problem, for which compressive sensing and sparse coding techniques are already providing many useful solutions.

MATERIAL APPEARANCE In this second part we have presented an intuitive control space for material appearance, which allows for artistic exploration of plausible material appearances based on perceptually-meaningful attributes. We have derived novel functionals connecting principal components of the BRDF to a high-level characterization of material appearance, inferred from a large-scale study with 400 participants. This characterization is made up of our appearance attributes, which are intuitive, descriptive, and discriminative with respect to many different reflectance properties, as we have shown. We have further analyzed the resulting appearance space, which has yielded insights on material perception, and proposed a number of example applications that can benefit from our approach.

Then, we have focused in one of this applications, and we have proposed a new two-step method for BRDF gamut mapping. In the first step we work in PC space, and use our previously proposed functionals that map this space to higher-level intuitive attributes to preserve the appearance of any of such attributes. The output of this first step, which only optimizes achromatic reflectance, is then used as input to an image-space optimization which brings the final BRDF into the ink gamut by expressing it as a convex combination of the available inks. We perform both an objective and subjective validation comparing against the state of the art. Additionally, we show how a slight modification of our framework can provide extended functionalities for intuitive material editing.

There are many opportunities for interesting future work. First, we do not claim to have found a complete, universal list of perceptual attributes defining appearance. This is an open problem, for which no established methodology exists. In fact, a key advantage of our flexible methodology is that it allows to define *custom* attributes, which may adapt better to a particular user or context, while avoiding mixed nomenclatures. Moreover, it can also be used on different databases. Second, it would be interesting to expand our approach to more materials; despite the fact that our extended MERL dataset provides a reasonably uniform coverage over a very wide range of isotropic appearances, some perceptual attributes can be under-represented. This translates into less user ratings, which may lead to less reliable functionals in some regions of our 5D space. Further, our system does not currently handle some complex appearance behavior such as color changes, grazing angle effects, or hazy gloss. These are undersampled in our dataset, and remain as future work, deserving further investigation.

VIRTUAL REALITY In the third part we have focused on virtual reality. First, we have collected a dataset that includes gaze and head orientation for users observing omnidirectional stereo panoramas in VR. The primary insights of our data analysis are: (1) gaze statistics and saliency in VR seem to be in good agreement with those of conventional displays; as a consequence, existing saliency predictors can be applied to VR using a few simple modifications described in this work; (2) head and gaze interaction are coupled in VR viewing conditions – we show that head orientation recorded by inertial sensors may be sufficient to predict saliency with reasonable accuracy without the need for costly eye trackers; (3) we can accurately predict time-dependent viewing behavior only within the first few seconds after being exposed to a new scene but not for longer periods of time due to the high inter-user variance; (4) the distribution of salient regions in the scene has a significant impact on how viewers explore a scene: the fewer salient regions, the faster user attention gets directed towards any of them and the more concentrated their attention is; (5) we observe two distinct viewing modes: attention and re-orientation, potentially distinguishable via head or gaze movement in real time and thus useful for interactive applications. These insights could have a direct impact on a range of common tasks in VR. We outline a number of applications, such as panorama thumbnail generation, panorama video synopsis, automatically placing cuts in VR video, and saliency-aware compression.

Many potential avenues of future work exist. Predicting gaze scanpaths of observers when freely exploring a VR panorama would be very interesting in many fields, including vision, cognition, and of course, any VR-related application. Also, our data can be of particular interest to build gaze predictors using just head movement as input, since head position is much cheaper to obtain than actual gaze data. Additionally, it would be interesting to explore how behavioral models could improve low-cost but imprecise gaze sensors, such as electrooculograms. Future work could also incorporate temporal consistency for saliency prediction in videos, or extend it to multimodal experiences that include audio.

Second, we have analyzed user behavior in more complex scenarios. Our work is the first to attempt a systematic analysis of viewer behavior and perceived continuity in narrative VR content. A systematic exploration of this topic is challenging for two main reasons: (1) the extreme high dimensionality of its parameter space; and (2) that it involves many discrete, categorical (as opposed to interval or ordinal) variables of influence. We have relied on the event segmentation theory, which has provided us with some solid ground to carry out our research, and have analyzed previous related studies on traditional cinematography. Our results may have direct implications in VR, informing content creators about the potential responses that certain edit configurations may elicit in the audience. For instance, for a fast-paced action movie our results suggest that regions of interest should be aligned across edits, while to evoke a more exploratory behavior, misalignments are recommended. As in all studies of similar nature, our results are only strictly valid for our chosen stimuli.

As future work, many other variables and parameters can be explored, such as other types of cinematographic cuts, longer movies, more complex visual content, the influence of sound, or the effect of fatigue or frequent exposure to VR content. More comprehensive subjective data may also be a valuable source of information, together with our objective gaze data.

Third, we have presented here a technique to enable head motion parallax in 360° video, thus enabling 6-DoF viewing of real-world capture footage. We have designed our method to be independent of a specific hardware, camera setup, or recorded baseline, showing examples from different common 360° capture systems, including depth estimated from scratch by a neural network. We assume that each RGB (and depth) frame is a monocular panorama in equirectangular projection, but other projections (such as cube map) are also possible. Our system requires only RGBD 360° video as input, and is rather robust to inaccuracies in the depth. Thus, it can deal with different data sources, from 360° video plus depth stitched and computed from individual videos from a camera rig, to 360° monocular video with depth computed with a CNN-based depth estimation algorithm. While having access to a multiview setup may not be commonplace, a number of 360° cameras do provide stereo output, which can yield a reasonable depth. Still some common cameras do not provide stereo (Ricoh Theta, Samsung Gear 360, or Nikon Keymission 360, among others), due, e.g., to the limited overlap between the camera views. Our method is designed to be agnostic to the hardware employed during capture, in order not to diminish generality. Our user studies confirm that our method provides a more compelling viewing experience, while reducing discomfort and sickness. Interestingly, the additional degrees of freedom enabled by our method also influence viewing behavior: On average, users displace more their head from the center of projection when viewing content with 6-DoF.

However, our method does still introduce some artifacts at disocclusion boundaries, which is to be expected given the extremely limited nature of our input. Our studies show that users prefer 6-DoF viewing to 3-DoF viewing, even with these artifacts; getting rid completely of such artifacts remains an open challenge. It is possible that combining ideas from our work and the works by Hedman et al. [134, 135] could lead to higher-quality 6-DoF capture. These artifacts increase as the viewer moves farther away from the center of projection. In the future, we would like to explore options to minimize this, for instance by creating a non-linear mapping between the head and the camera movement that prevents the viewer from moving too far. Alternatively, existing techniques for controling user attention in VR [74, 117] could be helpful in this context.

PERSONAL CONCLUSIONS This thesis has allowed me not only to improve my technical skills, but it has also provided me with the maturity as a researcher to tackle difficult problems and to adapt to different working environments. During these years I have learned to work in a team, and working in different institutions has helped me to understand and to adjust to the different working habits that different teams may have. Also, I have learned to seize the time I have, to be efficient, and that a good organization and planning saves a lot of time and prevents ugly bottlenecks. I have also come to terms with the fact that *things need to be done* and not everything can be always completely perfect as I would like it to be, that sometimes is good to step back and look at the bigger picture. Being able to supervise students has allowed me to significantly improve my communication skills, and it also has made me keep learning about the topics I was working on with them by trying to find different ways to explain more difficult concepts. Finally, I have to admit that when I started this thesis I was not sure that this was the path I wanted to follow, I was unsure of my abilities, and I was unsure about devoting so much time to it. After these years, I realize that I made the right choice, and that I have enjoyed every second (even the deadlines) that I have spent working on this thesis. All in all, it has been an invaluable experience.

10. Conclusiones

En esta tesis se han presentado una serie de contribuciones en tres áreas de la computación visual (visual computing): imagen computacional, apariencia de materiales, y realidad virtual.

La primera parte se ha enfocado a la adquisición eficiente de dos dimensiones la función plenoptica, lo que ha llevado a contribuciones en captura de video de alta velocidad, y captura de imágenes en alto rango dinámico.

En la segunda parte, se ha intentado abordar un problema persistente en gráficos, que es de la percepción y edición de materiales capturados. Para ello se ha propuesto un espacio de navegación basado en percepción que utiliza para describir los materiales atributos intuitivos de alto nivel. Se han derivado una serie de aplicaciones que demuestran la usabilidad de dicho espacio, incluyendo un interfaz de edición intuitiva de materiales, que ha demostrado ser más sencilla de usar con respecto a interfaces disponibles en software de modelado y edición. Además, se ha ahondado más en profundidad en una de estas aplicaciones, el mapeado de tono, para el que se ha propuesto un algoritmo basado en nuestro espacio de atributos perceptuales, y que mejora el estado del arte, disminuyendo el error introducido en el proceso de mapeado.

Finalmente, en la tercera parte, se han recorrido los primeros pasos hacia el entendimiento del comportamiento de usuarios y sus mecanismos atencionales en realidad virtual, comenzando primero con el estudio de escenas simples estáticas, y prosiguiendo con entornos dinámicos más complejos bajo una perspectiva cinematográfica. Además, hemos propuesto un sistema de visualización de vídeo 360 en realidad virtual que añade paralaje a vídeos originalmente planos, mejorando así la inmersión y la experiencia de visualización.

En concreto, las aportaciones de la tesis han sido las siguientes:

- Un sistema de reconstrucción de vídeo de alta velocidad basado en la captura de una sola imagen. Mediante técnicas de reconstrucción avanzadas el sistema desarrollado es capaz de reconstruir secuencias de vídeo a partir de una sola imagen con la información temporal codificada en la misma.

- Un sistema de captura de imágenes de alto rango dinámico en un único disparo. Este sistema permite usar cámaras comerciales para capturar escenas de alto rango dinámico de alta calidad en un solo disparo sin perder resolución, y sin modificaciones hardware adicionales. Además, este sistema se extiende naturalmente a vídeo, permitiendo así la captura de vídeos de alto rango dinámico.

- Un espacio de manipulación de apariencia de materiales basado en una serie de estudios formales de usuario. La manipulación de apariencia de materiales basada en atributos perceptuales es un problema muy complejo debido a la desconexión que existe entre los parámetros físicos y las características perceptuales de dichos materiales. El espacio propuesto ha resultado además de gran utilidad para una serie de aplicaciones que también han sido desarrolladas, incluyendo un sistema intuitivo de edición de materiales especialmente diseñado para usuarios noveles, y un algoritmo de mapeado de tono basado en atributos perceptuales.

- El primer análisis formal de exploración visual de escenarios panorámicos estáticos en sistemas de realidad virtual basado en estudios de usuario, que ha permitido identificar patrones en el comportamiento de usuarios clave para el desarrollo de aplicaciones como, por ejemplo, predicción de zonas de atención, o compresión de panoramas para su posterior visualización en realidad virtual sin pérdida de calidad percibida.

- El primer análisis formal de continuidad cinematográfica y comportamiento de usuarios en sistemas de realidad virtual basado en estudios de usuario. Las conclusiones derivadas del estudio aportan una serie de pautas para la creación y edición de contenido cinematográfico para realidad virtual.

- Un algoritmo capaz de añadir paralaje a videos de 360 grados capturados (tanto estereoscópicos como monoculares) para su reproducción en dispositivos de realidad virtual. Este algoritmo permite que vídeos que habían sido capturados desde un único punto de vista, puedan ser visualizados desde puntos de vista ligeramente diferentes. Los resultados de este trabajo permiten una visualización más natural y cómoda de contenido de vídeo de estas características en realidad virtual.

Part VI

Appendices

A. Convolutional sparse coding for high-speed video acquisition: additional data

A. 1 Database

Our database provides scenes of different nature, and a wide variety of spatial and temporal features. Representative frames of the videos in the database are shown in Figure A.1.

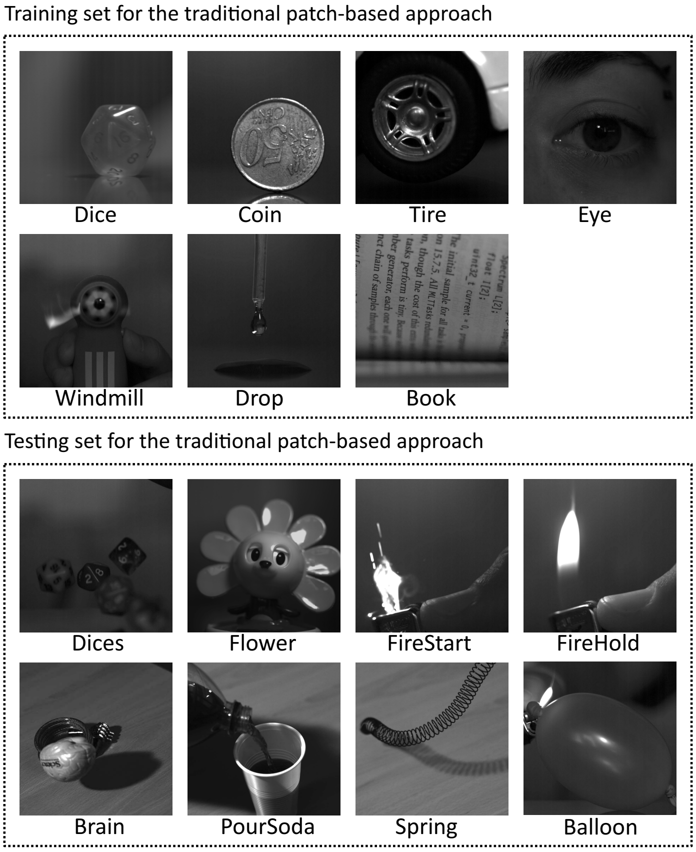

Training set for the traditional patch-based approach
Dice
Coin
Tire
Eye
Windmill
Drop
Book
Testing set for the traditional patch-based approach
Dices
Flower
FireStart
FireHold
Brain
PourSoda
Spring
Balloon

Figure A.1: Representative frames of the high-speed videos (1000 fps) in our database.

A. 2 Patch-based compressive acquisition of high-speed video

Here we explain in detail the pipeline of a system based on compressive sensing applied to high speed video capture, including the reconstruction of a video from a single coded image, and the process of building a dictionary appropriate for high speed video representation, using a patch-based approach. The pipeline is presented in Figure A.2.

A.2.1 *Learning high-speed video dictionaries*

We need a dictionary in which the signals of interest, in this case high speed videos, are sparse. The advantage of choosing an already existing basis is that usually these bases are mathematically

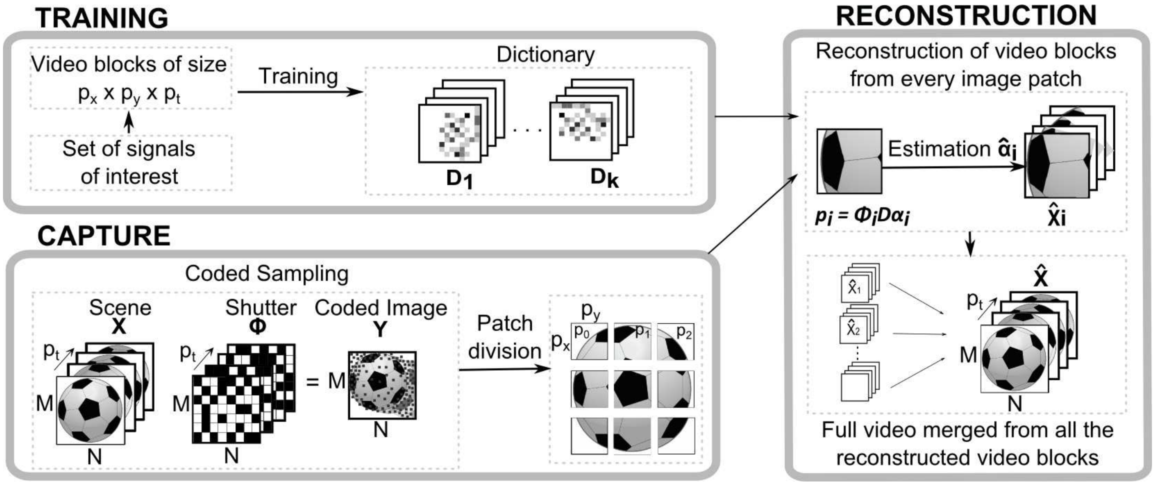


Figure A.2: Pipeline of the system for high speed video acquisition and reconstruction with the traditional patch-based approach. Top left: Training of a dictionary D with k elements with a set of video blocks of size $p_x \times p_y \times p_t$ from the collection of signals of interest. Bottom left: Coded sampling of the original scene X with a measurement matrix ϕ resulting in the coded image Y, and division of this image in patches p_i appropriate for the size of the blocks used for training the dictionary. Right: Reconstruction of the video blocks $\hat{X}_i$ from each image patch p_i and merging of the blocks to obtain the full video $\hat{X}$.

well defined and their properties are known. However, since they are designed to be generic, they usually do not provide an optimal sparse representation for a specific set of signals of interest. A way to ensure that our set will be sparse in the basis domain is to train a dictionary specifically adapted for sparse video representation. We learn the fundamental building blocks (atoms) from high speed videos in our database (see Section A.1), creating an overcomplete dictionary. For training, we use the DLMRI-Lab implementation [264] of the K-SVD [4] algorithm, which has been widely used in the compressive sensing literature.

From the database videos we have obtained a training set by splitting some of these videos into blocks of size $n = p_x \times p_y \times p_t$. Given a large collection of blocks, we have to choose a computationally affordable number of blocks as a training set. Most of these blocks will not have interesting features such as gradients or temporal events, therefore we would like to discard some of them while ensuring the presence of blocks with relevant information. In this work we propose an alternative to random selection that enforces this by giving certain priority to blocks with high variance. This is further explained in Section A.3, where we also analyze the alternative choice of an existing base (DCT Type-II) instead of a trained dictionary.

A.2.2 Capturing coded images from high-speed sequences

The measurement matrix, for the particular case of video sampling, consists on a coded exposure implemented as a shutter function that samples different time instants for every pixel. The final image is thus formed as the integral of the light arriving to the sensor for all the temporal instants sampled with the shutter function. This can be expressed with the following equation:

$$I(x,y) = \sum_{t=1}^{T} H(x,y,t)A(x,y,t) \tag{A.1}$$

where $I(x,y)$ is the captured image, H the shutter function and A the original scene. In a conventional capture system $H(x,y,t) = 1 \;\forall\; x,y,t$ but in this case the goal is to achieve a H function that fulfils the properties of a measurement matrix suitable for compressive sensing reconstruction. An easy way to fulfil this requirements is to build a random sampling matrix. However,

a fully-random sampling matrix cannot be implemented in current hardware, as explained below. Therefore, we use the shutter function proposed by Liu et al. [141, 195] that tries to achieve randomness while imposing some restrictions to make a hardware implementation possible.
Hardware Limitations Image sensors are usually based on CMOS technology and a capture process comprises several processes requiring a certain amount of time, making difficult to build a function that samples randomly every pixel at every time instant. There are two key limitations: maximum integration time, i.e, the maximum time we can sample a pixel, which is limited by thermal noise in the sensor; and speed at which the shutter can be opened and closed, which is limited by the processing time of the camera (such as A-D conversion and readout time). Because of this, the shutter design for each pixel is limited to a single continuous exposure and this exposure time has to be shorter than the integration time of the camera, that is, the shutter function design is limited to a single continuous integration bump for each pixel and this bump also has a limited duration. This function can be easily implemented in a *DMD* or an *LCoS* placed before the sensor, as proven by Liu et al. [141, 195].

A.2.3 *Reconstructing high-speed videos from coded images*

Once the dictionary and the measurement matrix are decided, we need to solve an optimization to estimate the α coefficients and thus be able to reconstruct the signal. Many algorithms have been developed for solving this minimization problem for compressive sensing reconstruction. In Section A.3 we analyze the influence of this algorithm in the quality of the results by comparing two algorithms that had proven a good performance in similar problems: Orthogonal Matching Pursuit (OMP) [248] and the LARS approximation for solving the Lasso [91]. We use the implementations available in the SPArse Modeling Software (SPAMS) [203].

A3 Additional analysis

A.3.1 *Convolutional approach*

In this section we analyze the size of the trained filters for the convolutional approach and its influence in the quality of the results. We train dictionaries of 100 filters and variable sizes, and we calculate the PSNR of the reconstructions over our set of testing videos. We show in Table A.1 the average PSNR for the different sizes tested. We have found the reconstruction quality to be robust for different sizes due to the convolutional nature of the algorithm.

Table A.1: Average PSNR for our set of testing videos, and for different filter sizes.

Filter size (pixels)	**PSNR (dBs)**
7 x 7	22.51
9 x 9	22.56
11 x 11	22.57
13 x 13	22.52

A.3.2 *Patch-based approach*

In this section we include further analysis of the patch-based system presented in the main text, exploring the parameters of influence. All the learned dictionaries presented in this section are trained with the same set of videos (see Figure A.1). We analyze several parameters over a test set of six videos (see Figure A.3) to find the parameter combination yielding the best results. Note that none of the testing videos are used during the training. In order to isolate the influence of each parameter in the framework, we maintain the rest of the parameters fixed while we vary the one being analyzed. The set of parameters derived from this analysis is used to obtain the results for reconstructions based on the patch-based approach. We use as measures of quality the PSNR (Peak Signal to Noise Ratio), widely used in the signal processing literature. We also performed all tests with the MS-SSIM metric [335], which takes into account visual perception, and found that it yielded results consistent with PSNR. Thus, for brevity, we only show results with PSNR.

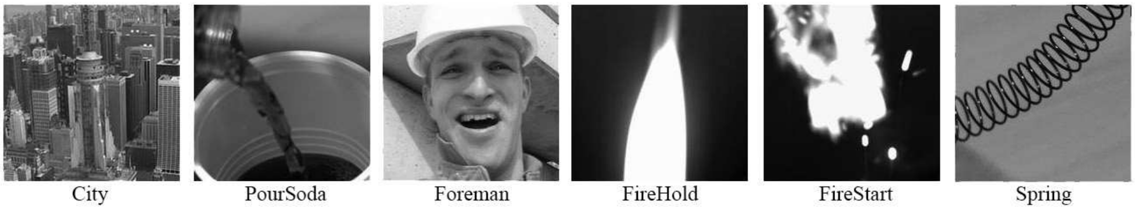


Figure A.3: Sample frame extracted from each of the videos used in the analysis.

CHOOSING A DICTIONARY Here we compare our trained dictionary with a three dimensional DCT basis under the same conditions, i.e., same number of elements of the dictionary, and same block size. We can see in Figure A.4 that the training dictionary performs better than the DCT basis except for the case of the video *Spring*. This video is consistently the result with worst quality, so the better performance of DCT can be arbitrary or due to different amounts of noise and artifacts in the reconstruction. Therefore, we choose the trained dictionary as the best alternative.

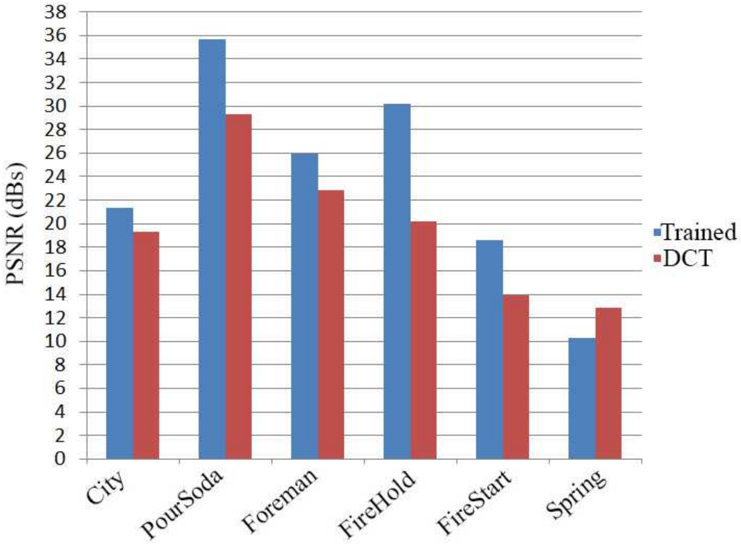


Figure A.4: Quality of the reconstructed videos (in terms of PSNR) for the set of analyzed videos reconstructed with a trained dictionary and with a three dimensional DCT basis under the same conditions (same dictionary elements and same block size). The trained dictionary consistently outperforms DCT, except for the *Spring* video, which can be considered an outlier (see text for details).

SIZE OF TRAINING BLOCKS We use for the training a high speed video dataset and we divide each video into blocks. The size of these blocks is the size of the atoms of the resulting trained dictionary as well as the size of the video blocks the reconstruction is able to recover. We now analyze in Figures A.5 and A.6 the impact of the spatial and temporal size of the blocks in the quality of the results as well as in the reconstruction times.

For the spatial size of the blocks, we see a consistent improvement of the quality in the results if we increase the size from 5*x*5 to 7*x*7 pixels, however if we keep increasing this size up to 9*x*9 pixels there is not a clear improvement. Aditionally, for the later the average time for the reconstruction increases drastically, so we chose a spatial size of 7*x*7 for our reconstructions. Note that it is important that each block is large enough to contain significant features (such as edges or corners), but not too large in order to avoid learning very specific features of the training videos (and thus overfitting).

For the temporal size of the blocks, we observe the quality decreases when we increase the temporal dimension of the blocks. This is to be expected since as the temporal size increases, more information (number of frames) needs to be coded within the same number of pixels (a single image). On the other hand, the reconstruction time increases almost linearly. Therefore, choosing the temporal size of the blocks is a tradeoff between quality and the number of frames to be coded (and thus the maximum achievable temporal resolution). We chose 20 frames in order to be able to code significant temporal variations in our scenes.

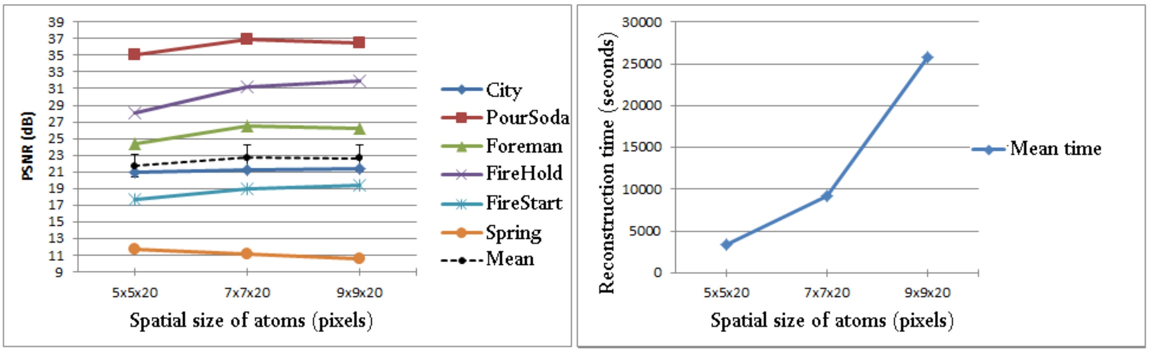


Figure A.5: Quality (in terms of PSNR) and times of the reconstruction for blocks of different spatial sizes for each of the analyzed videos.

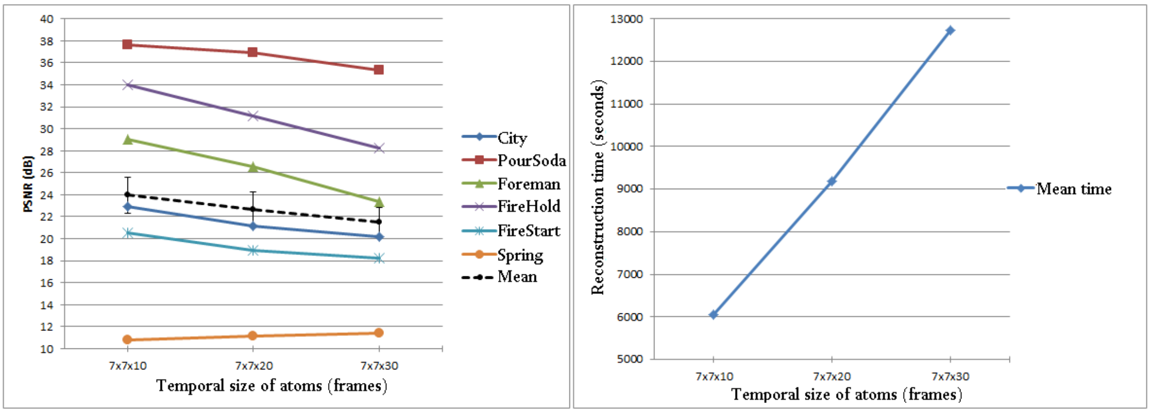


Figure A.6: Quality (in terms of PSNR) and times of the reconstruction for blocks of different temporal sizes for each of the analyzed videos.

RECONSTRUCTION ALGORITHM As explained in Section A.2, we need a reconstruction algorithm to recover the **s** coefficients that are multiplied by the dictionary to obtain the original scene from the coded image; this is posed as a minimization problem. This algorithm is used in the reconstruction as well as in the training, since the training algorithm K-SVD solves the minimization problem as a step to update the dictionary. In this work, we analyze two algorithms for solving the minimization: Orthogonal Matching Pursuit [323] and the LARS solver for the Lasso [91]. In

Figure A.7 we show combinations of these two algorithms in the training and reconstruction steps. It can be seen that the reconstruction algorithm has a significant influence in the reconstruction step, with LARS-Lasso yielding the best performance.

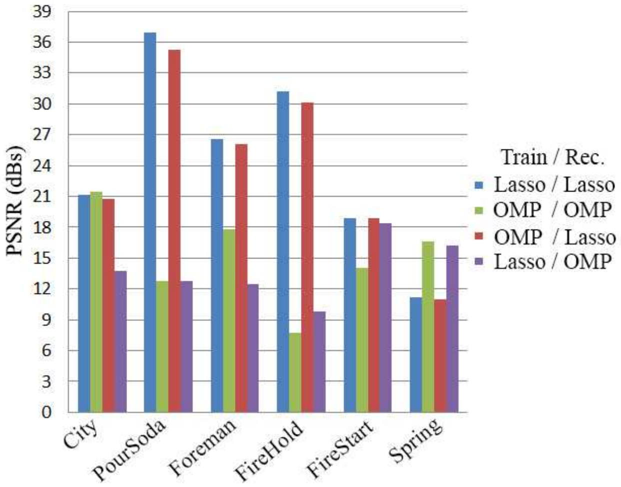


Figure A.7: Quality of the reconstruction (in terms of PSNR) for different combinations of the algorithms OMP and LARS-Lasso for the training/reconstruction steps for each of the analyzed videos. The best combination is obtained using the LARS-Lasso both for training and reconstruction.

MEASUREMENT MATRIX Choosing the best measurement matrix possible is crucial to achieve good results in a compressive sensing framework. As explained in the main text, some properties need to be fulfilled by this matrix and at the same time we want it to be implementable in hardware. The main property it has to fulfil is incoherence. In order to satisfy the incoherence condition, it is necessary to sample the signal with a particular pattern that guarantees incoherence of such pattern with the chosen basis. The coherence between two pairs of bases measures the largest correlation between any two elements of those bases. For this purpose it has been demonstrated that a random sampling, in particular for Gaussian and binary distributions, yields a good amount of incoherence in an overall system formed by the sampling pattern and any basis. Here we compare several measurement matrices [141, 195] (see Figure A.8):

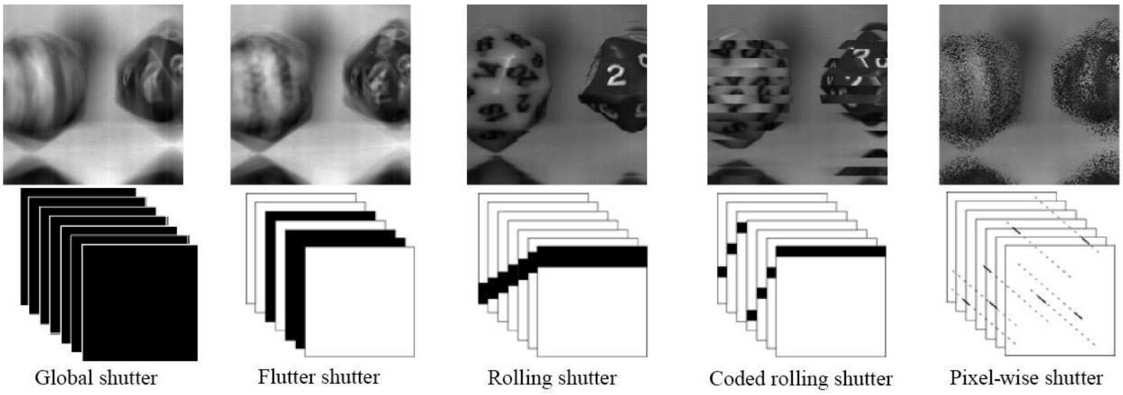


Figure A.8: Shutter functions and resulting coded images when sampling with them. Black pixels correspond to pixels being sampled (on) while white pixels correspond to pixels that are off.

- **Global shutter**: All the pixels are sampled over the integration time of the camera.

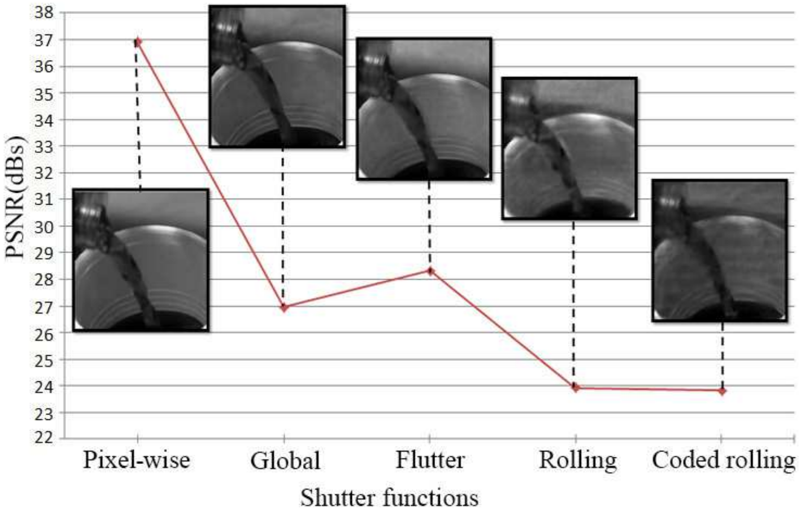


Figure A.9: Quality of the reconstruction (in terms of PSNR) using several measurement matrices. The insets show a reconstructed frame for every measurement matrix.

- **Flutter shutter** [261]: The shutter entirely opens and closes over the integration time of the camera.
- **Rolling shutter**: Pixels are sampled sequentially by rows through time.
- **Coded rolling shutter** [120]: Variant of the rolling shutter. In this case the image is sub-divided and each part sampled with rolling shutter independently. For example, for an image sub-divided in two, first odd rows are sampled with rolling shutter, and then even rows.
- **Pixel-wise shutter** [141, 195]: Each pixel is sampled over a fixed bump time shorter than the integration time of the camera, starting at different time instants. However, in order to obtain enough samples for the reconstruction a condition is imposed: Taking into account the size of the blocks we want to reconstruct, for every temporal instant at least one of the pixels from each block must be sampled. This is a way to ensure that every temporal instant is represented in every patch of the captured image. This can be expressed as:

  $$X = \sum_{(x,y)\in p_j} H(x,y,t) \geq 1 \; for \; t = 1..T \tag{A.2}$$

  with T the temporal instants (or frames) to sample.

The results of the analysis of these measurement matrices are shown in Figure A.9. The best results are obtained with the pixel-wise shutter, since it is specifically designed for the compressive sensing framework, as it guarantees that at least one pixel per patch is sampled for every frame.

B. An intuitive control space for material appearance: additional details and analysis

B.1 additional details on experiments

B.1.1 *Experiment 0: Principal components space*

To make sure that choosing only five principal components does not affect the perception of appearance, we have run a pilot test. We use as stimuli 20 BRDFs from the MERL database, manually selected to cover a wide range of appearances. The test follows the two-alternative forced choice (2AFC) methodology. Two images are presented to the user: the original BRDF, next to the same BRDF represented with only the first five components. Each comparison asks about a particular attribute from our list. The user has to choose which of the two depictions of the material better conveys such attribute (for instance, *Which of the two images looks more metallic?*). The order and relative location of each version was randomized. Each subject was shown 25 BRDFs, and a total of 47 subjects took part in the experiment. On average, the BRDF represented with five components was chosen 49% of the times. We ran χ^2 tests per attribute to confirm whether users where actually picking randomly between both options, results are presented in Table B.1. For all attributes we obtained a very high p-value, which speaks highly in favor of a random selection by subjects.

We conclude that a five dimensional space suffices for our subsequent tests. Additionally, limiting the space to five dimensions has an additional advantage: When sampling the space to create a larger database of BRDFs, the reduction of the space to 5D improves the sampling process by avoiding the placement of samples in regions of the space with little impact on appearance.

Table B.1: Results of the χ^2 test for the principal component space experiment.

	χ^2	**Df**	**p-value**
Plastic-like	0.0169	1	0.8965
Rubber-like	0.3137	1	05754
Metallic-like	0.0041	1	0.9489
Fabric-like	0.6792	1	0.4098
Ceramic-like	0.4010	1	0.5265
Matte	0.0051	1	0.9430
Glossy	0.0727	1	0.7874
Bright	0.6545	1	0.4185
Rough	0.2426	1	0.6223
Strength of reflections	0.7059	1	0.4008
Sharpness of reflections	0.4175	1	0.5181

B.1.2 *Experiment 1: Building the space of attributes*

Finding a parameter space providing an intuitive representation of material appearance is a long-standing problem, for which no definite answer [63, 97] nor methodology [280] exist, whereas usually naming depends on the field [1]. The parameter space must be reduced enough to be manageable, but also comprehensive enough to allow for rich yet intuitive appearance edits, even for inexperienced users. For this first test, we rendered a large number of stimuli depicting different materials, built an extensive initial list of candidate appearance descriptors, and then relied on a user study to reduce them to a suitable size. We included in our list attributes ranging from high level class descriptors (e.g. ceramic-like) to low level appearance descriptors (e.g., strength of reflections). Relying on Fleming's work [101], where he states that *we can also make many judgments about the perceived qualities of different materials irrespective of their class membership*, we do not make any restrictions about the type of descriptors in our list.

STIMULI Inspired by recent works on material perception and design (e.g., [152, 164, 233]), our stimuli consist of spheres of 60 different materials from the MERL database [219], chosen to span a wide range of different appearances. The spheres are lit by direct illumination. We render them using PBRT, and the *St. Peter's* environment map from the Light Probe Image Gallery [76], since real-world illumination, and that environment map in particular, facilitates material perception in single images [102].

INITIAL LIST OF ATTRIBUTES We compiled an extensive list of appearance attributes from previous works in industry and academia [41, 146, 337, 343]. Additionally, seven subjects were asked to provide, for each of our 60 stimuli, at least four attributes that described its appearance, using their own words; this yielded a second initial list of attributes. We ensured that each stimulus was seen by at least two people. We then joined the two lists and reduced the number of entries by clustering semantically equivalent attributes; from this we obtained our initial list of 28 appearance attributes (see Sec. H).

PARTICIPANTS Twenty-six paid subjects took part in our experiment, under controlled conditions in our lab. They all had self-reported normal or corrected-to-normal vision, and had no graphics background.

PROCEDURE We seek to further reduce the initial 28 attributes, keeping only those meaningful and understandable even by inexperienced users, and reasonably well represented in our database. To do this, we devised an experiment in which subjects had to establish, for each stimulus shown, whether each of the attributes applied to the material or not. Each subject was randomly shown 12 stimuli on a calibrated display, and there was no time limit (on average the complete tests took around 20 minutes per subject). Among the stimuli, a specific BRDF (the same for all subjects) was shown twice throughout the experiment, and served as a control stimulus for outlier rejection. This experiment would tell us: First, for which attributes there is a high agreement between users, and therefore they are clearly understood; and second, which attributes systematically received negative answers and thus are not representative of material appearance in our database.

ANALYSIS AND MAIN FINDINGS We first computed agreement, as the percentage of responses coincident with the majority answer. Additionally, we computed Hamming distances between answers for different attributes, as an indicator of correlation between them, and confirmed these correlations using Pearson's chi-square test [100, 250], which analyzes whether there is a relationship between each pair of attributes, as well as the strength of this relationship. Attributes were then removed according to three conditions: a chi-square value above 65, an agreement below 0.8, or a Hamming distance below 0.2. The final list consists of fourteen attributes, covering both high- and mid-level features: *plastic-like, rubber-like, metallic-like, fabric-like, ceramic-like, soft, hard, matte, glossy, bright, rough, tint of reflections, strength of reflections*, and *sharpness of reflections*.

B.1.3 *Data pre-processing: Outlier rejection*

In Experiment 2 we gather up to 56,000 responses from 400 subjects via Amazon Mechanical Turk. Since these responses are perceptual ratings which we will use to derive our mappings (*attribute-PC space*), we need an effective outlier rejection prior to using the gathered data to fit the RBFNs.

We use the BRDF shown in Figure B.1 as a control question to reject outliers. We discard full subjects that do not have a reasonable answer to very clear attributes regarding our control image. We discard users that rate the control BRDF as very glossy (*Glossy* = 4 or 5), very metallic (*Metallic* = 4 or 5), with very strong reflections (*Strength of reflections* = 4 or 5), or with very sharp reflections (*Sharpness of reflections* = 4 or 5).

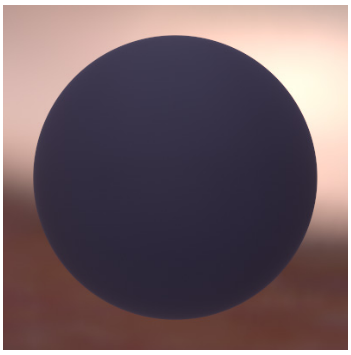
We also discard BRDFs from our experiment that are confusing for most of the users. We do this by calculating the difference between the 3rd and the 1st percentile of the observations. If this difference is greater than two for more than four attributes of the BRDF, we consider this BRDF as confusing for the users.

Finally, we discard outliers regarding observations for each attribute in each brdf. We do this if the observations fulfills any of the following conditions:

$$Observation < (P_1 - K_d * P_d) \text{ or } Observation > (P_3 + K_d * P_d) \tag{B.1}$$

with $P_d = P_3 - P_1$ and $Kd = 1.5$.

Figure B.1: Control image used for outlier rejection in our experiments



B.1.4 *User study interface*

We show in Figure B.2 the web-based interface used for the Experiment 0 (2AFC), and in Figure B.3 the web-based interface used for the Experiments 1 and 2 (Likert rating).

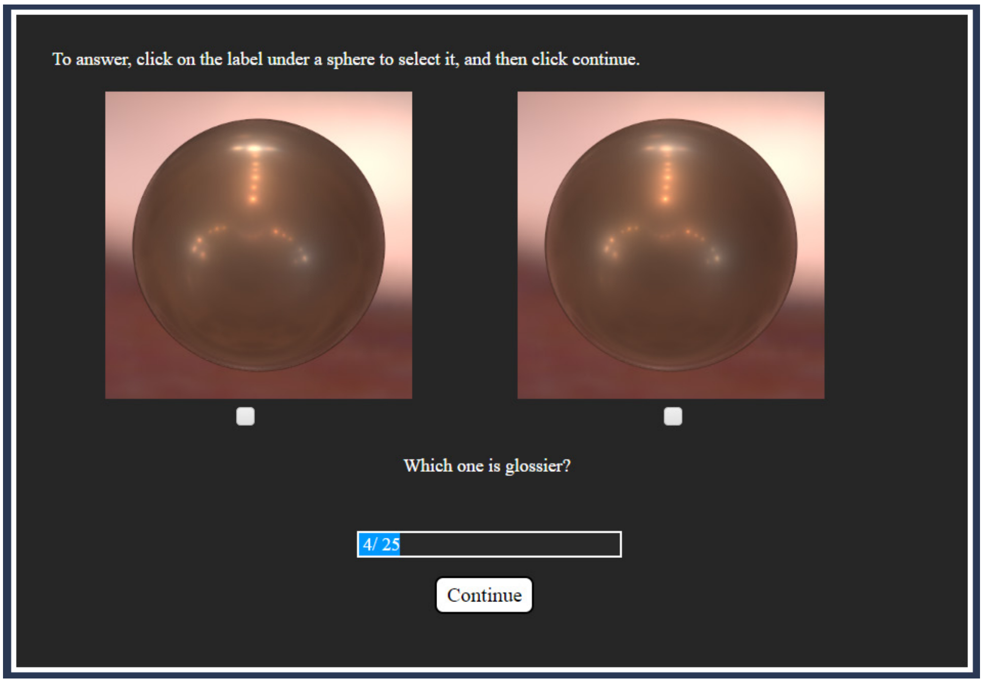
B.2 Per cluster analysis

In Figures B.4 and B.5 we show a per-cluster analysis of the mean and variance. Please refer to the main text for cues on how to interpret these plots.

B.3 Proof of concept with novice users

We provide in Figs. B.6, B.7, and B.8 additional results of the proof of concept test described in the main text.

Figure B.2: Web-based interface used for the experiment 0

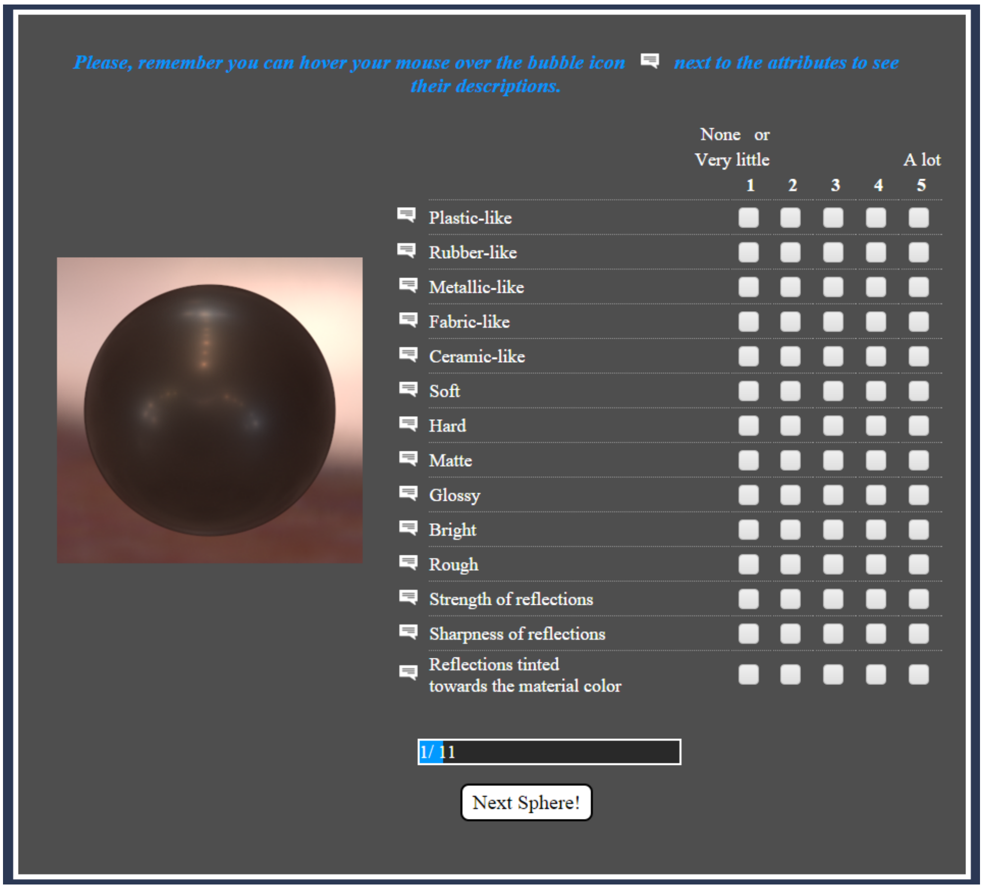


To answer, click on the label under a sphere to select it, and then click continue.

□ □

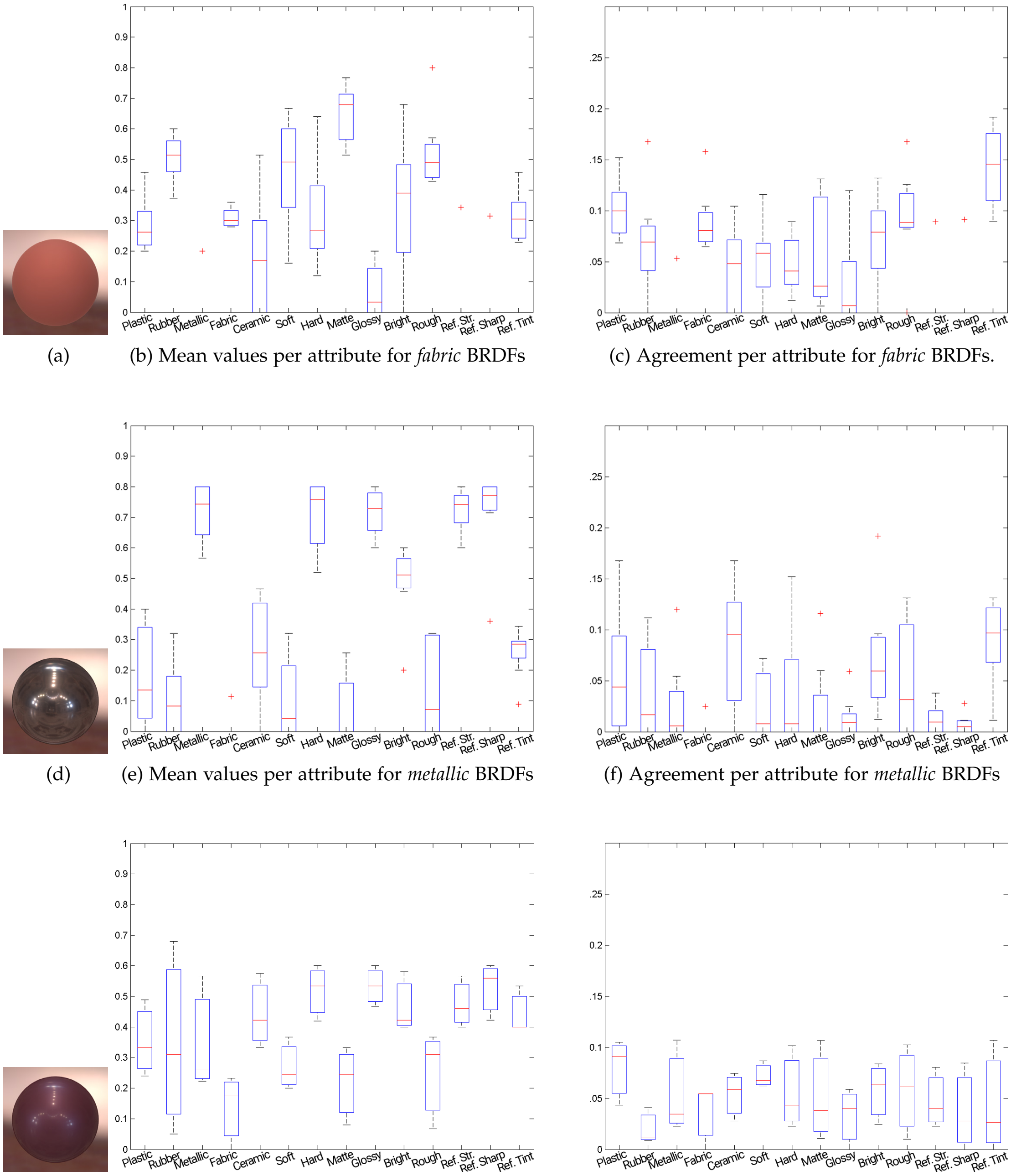
Which one is glossier?

Figure B.3: Web-based interface used for the experiments 1 and 2



Please, remember you can hover your mouse over the bubble icon next to the attributes to see their descriptions.

	None or Very little 1	2	3	4	A lot 5
Plastic-like	□	□	□	□	□
Rubber-like	□	□	□	□	□
Metallic-like	□	□	□	□	□
Fabric-like	□	□	□	□	□
Ceramic-like	□	□	□	□	□
Soft	□	□	□	□	□
Hard	□	□	□	□	□
Matte	□	□	□	□	□
Glossy	□	□	□	□	□
Bright	□	□	□	□	□
Rough	□	□	□	□	□
Strength of reflections	□	□	□	□	□
Sharpness of reflections	□	□	□	□	□
Reflections tinted towards the material color	□	□	□	□	□

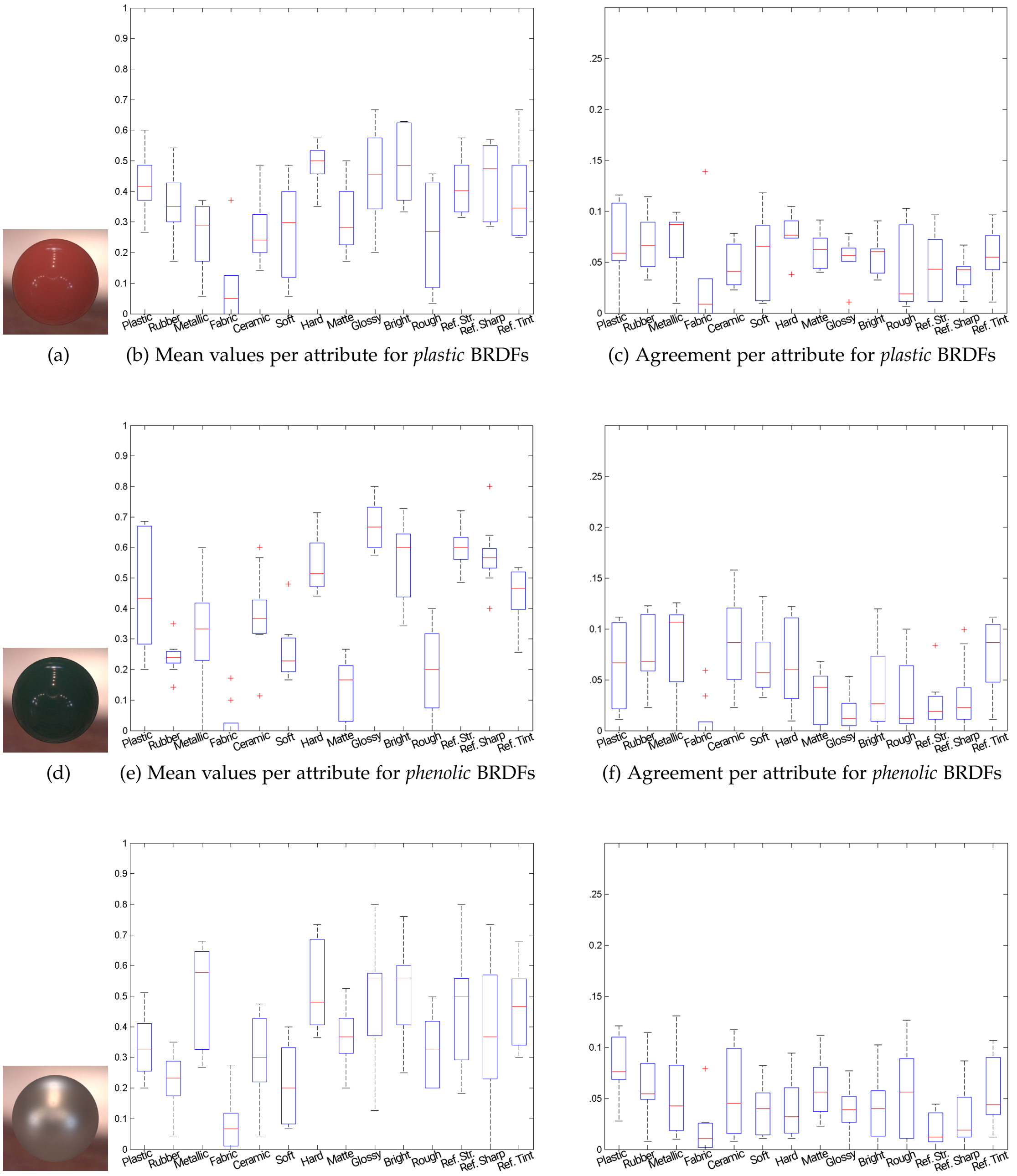


(a) (b) Mean values per attribute for *fabric* BRDFs (c) Agreement per attribute for *fabric* BRDFs.

(d) (e) Mean values per attribute for *metallic* BRDFs (f) Agreement per attribute for *metallic* BRDFs

(g) (h) Mean values per attribute for *acrylic* BRDFs (i) Agreement per attribute for *acrylic* BRDFs

Figure B.4: Means and variances for different types of BRDFs (I).



(a) (b) Mean values per attribute for *plastic* BRDFs (c) Agreement per attribute for *plastic* BRDFs

(d) (e) Mean values per attribute for *phenolic* BRDFs (f) Agreement per attribute for *phenolic* BRDFs

(g) (h) Mean values per attribute for *metallic-paint* BRDFs (i) Agreement per attribute for *metallic-paint* BRDFs

Figure B.5: Means and variances for different types of BRDFs (II).

Figure B.6: Results from editing the BRDFs *Pair* #1. The task was performed by three different novice users and consisted on finding a BRDF of intermediate appearance given an initial and a final appearance, with 3ds Max (bottom row) and our prototype (top row). Our prototype yields more similar results across users, and allows them to achieve better results in less time.

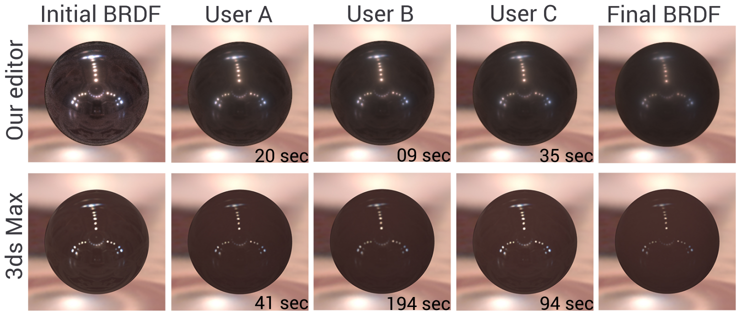


Figure B.7: Results from editing the BRDFs *Pair* #2. The task was performed by three different novice users and consisted on finding a BRDF of intermediate appearance given an initial and a final appearance, with 3ds Max (bottom row) and our prototype (top row). Our prototype yields more similar results across users, and allows them to achieve better results in less time.

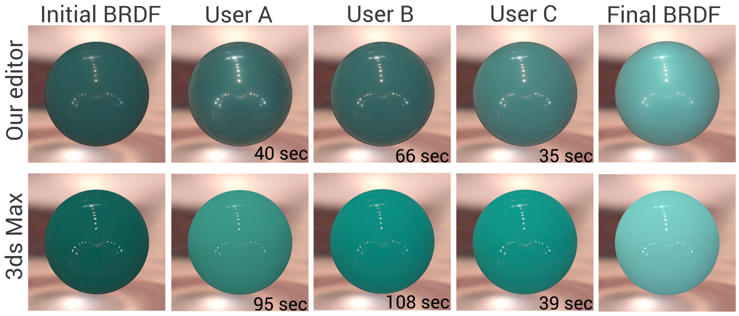
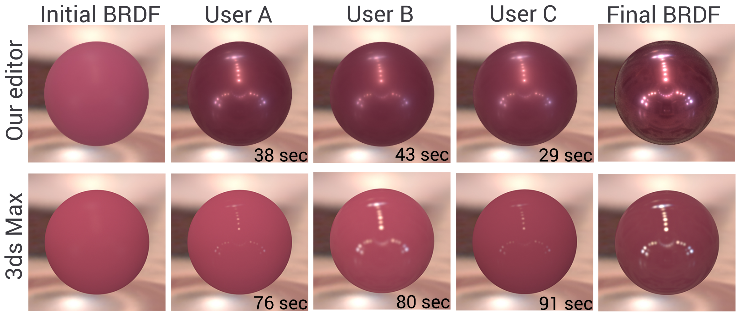


Figure B.8: Results from editing the BRDFs *Pair* #3. The task was performed by three different novice users and consisted on finding a BRDF of intermediate appearance given an initial and a final appearance, with 3ds Max (bottom row) and our prototype (top row). Our prototype yields more similar results across users, and allows them to achieve better results in less time.



B. 4 Attribute lists

We compiled an extensive list of appearance attributes from previous works in industry and academia. Additionally, seven subjects were asked to provide, for each of our 60 stimuli, at least four attributes that described its appearance, using their own words. We then joined the two lists and reduced the number of entries by clustering semantically equivalent attributes; from this we obtained the following initial list of 28 appearance attributes: *Plastic-like, Rubber-like, Mirror-like, Metallic-like, Ceramic-like, Fabric-like, Acrylic-like, Pearlescent, Velvety, Organic, Golden, Silver, Polished, Varnished, Chromed, Coated, Opaque, Soft, Matte, Shiny, Rough, Strength of reflections, Sharpness of reflections, Tint of the Specular, Sheen, Tint of the sheen, Haze, Specular Gloss*

The initial list of attributes was reduced to be manageable. To do this, we devised an experiment in which subjects had to establish, for each stimulus shown, whether each of the attributes applied to the material or not. The outcome of this experiment (Exp. 1 described in Sec. A.2 in this document) was the following list of perceptual attributes: *Plastic-like, Rubber-like, Metallic-like, Fabric-like, Ceramic-like, Soft, Hard, Matte,Glossy, Bright, Rough, Tint of reflections, Strength of reflections, Sharpness of reflections*

C. Saliency in VR, how do people explore virtual environments?: Additional analysis and data

C. 1 Recording head orientation and gaze

C.1.1 *Procedure*

All VR scenes were displayed using an Oculus DK2 head-mounted display, equipped with a pupil-labs stereoscopic eye tracker recording at 120 Hz. The DK2 offers a field of view of $95 \times 106°$. Prior to displaying test scenes, the eye tracker was calibrated using a per-user, per-use procedure. The Unity game engine was used to display all scenes and record head orientation while the eye tracker collected gaze data on a separate computer. Users were instructed to freely explore the scene and were provided with a pair of earmuffs to avoid auditory interference in the visual task. To guarantee consistency in the starting condition for different users, a gray environment with a small red box was displayed between different scenes, and users were instructed to find it. 500 ms after they aligned their head direction with the red box, a new scene would appear. Each panorama was displayed for 30 seconds. Scenes and starting points were randomized, while ensuring that each user would only see the same scene once from a single random starting point. Each user was shown 8 scenes; the total time per user of the experiment, including calibration, was approximately 10 minutes.

For the desktop condition, users sat 0.45 meters away from a 17.3″ monitor with a resolution of 1920×1080 px, covering a field of view of $23 \times 13°$. We used a Tobii EyeX eye tracker with an accuracy of 0.6° at a sampling frequency of 55 Hz [110]. The image viewer displayed a rectilinear projection of a $97 \times 65°$ viewport of the panorama, typical for desktop panorama viewers on the web. We calibrated the eye tracker and instructed the users on how to use the image viewer, before showing the 22 scenes for 30 seconds each. We paused the study after half of the scenes to recalibrate the eye tracker. In this condition, we only collected gaze data but not head orientation because the field of view is much smaller and users rarely re-orient their head. Instead, we recorded where the users interactively place the virtual camera in the panorama as a proxy for head orientation.

C.1.2 *Data processing*

PROCESSING EYE TRACKER SAMPLES For the VR setup, we discard and linearly interpolate recorded scan paths when the eye tracker reported low confidence values (i.e., < 0.7). The gaze tracker and the inertial measurement unit (IMU) record at different sampling rates. We resample the IMU to match the sampling rate of the gaze tracker (i.e., 120 Hz). For the desktop condition, we also discard and interpolate scan path measurements with low confidence. For both conditions, we calculated head/camera and gaze speeds, as well as accelerations, as the first and second derivatives of the latitude and longitude using the forward finite difference method. Before computing head and gaze statistics, we manually identified and filtered outliers at least 5 standard deviations away from the mean, accounting for less than 0.5% of the measurements.

FIXATIONS AND SALIENCY MAPS To identify fixations, we transformed the normalized gaze tracker coordinates to latitude and longitude in the 360° panorama. This is necessary to detect users fixating on panorama features while turning their head. We used thresholding based on dispersion and duration of the fixations [272]. For the VR experiments, we set the minimum duration to 150 ms [272] and the maximum dispersion to 1° [30]. For the desktop condition, the

Tobii EyeX eye tracker output showed significant jitter, and visual inspection revealed that this led to skipped fixations. We thus smoothed this data with a running average of 2 samples and detected fixations with a dispersion of 2° after the interpolation described above. Subsequently, when we compare the *VR* and the *desktop* conditions, we use for computing the continuous saliency map a gaussian blur with $\sigma = 2°$. We counted the number of fixations at each pixel location in the panorama. Similar to Judd et al. [158], we only consider measurements from the moment where user's gaze left the initial starting point to avoid adding trivial information. We convolved these fixation maps with a Gaussian with a standard deviation of 1° of visual angle to yield continuous saliency maps [181].

C. 2 Understanding viewing behavior in VR

C.2.1 *Is viewing behavior similar between users?*

We show in Figure C.1 the average ROC for all the 22 scenes for the *VR seated* (left) and *desktop* (right) conditions. The fast convergence of these curves to the maximum rate of 1 indicates a strong agreement, and thus similar behavior, between users for each of the scenes tested.

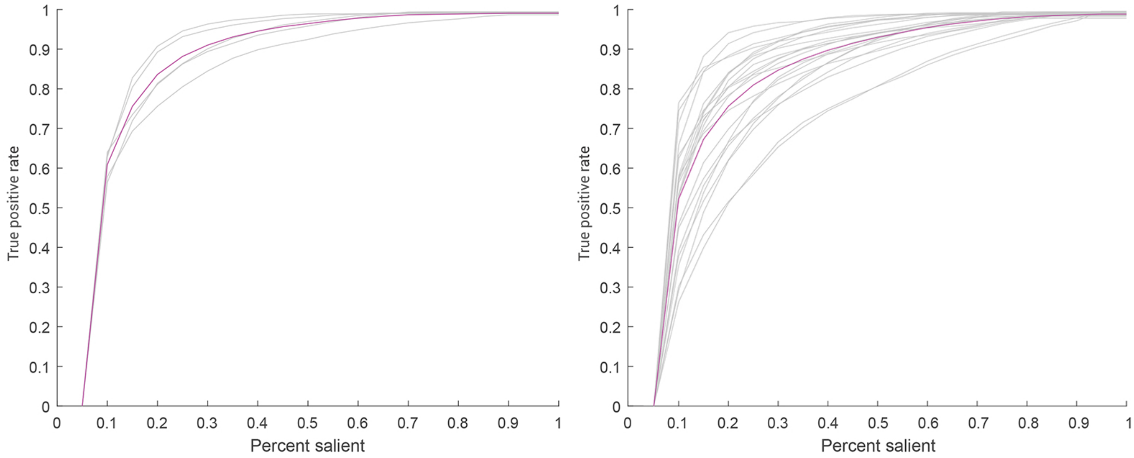


Figure C.1: ROC curve of human performance averaged across users (magenta)and individual ROCs for each scene (light gray) for the *VR seated* (left) and *desktop* (right) conditions. The fast convergence to the maximum is indicative of a strong agreement between users.

C.2.2 *Is there a fixation bias in VR?*

In Figure C.2, we show the average saliency map for the data on the five scenes that we captured for the sitting condition. Though only five of the 22 scenes were captured, the equator bias is clearly visible.

C.2.3 *How are head and gaze statistics related?*

In Figure C.3, we show the exploration progress for the sitting condition. The exploration progress is the average time that users took to move their eyes to a certain longitude relative to their starting point. On average, users fully explored each scene after about 17 seconds.

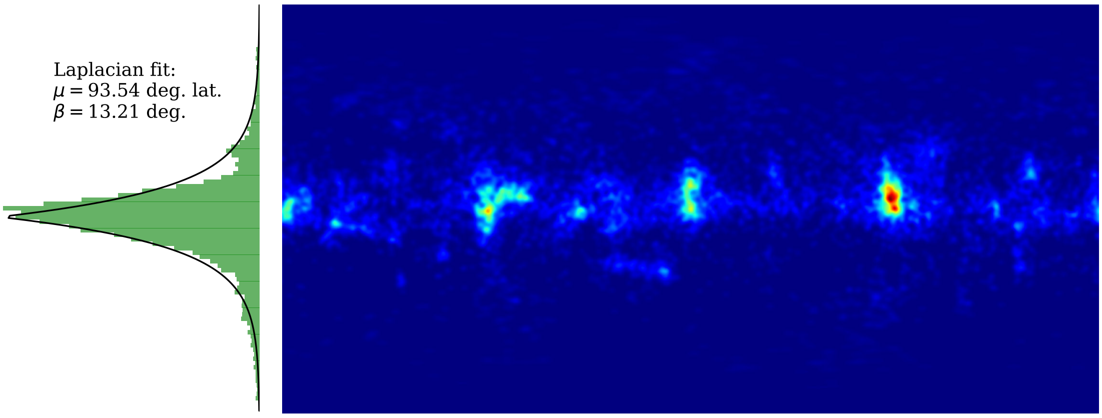


Figure C.2: Average saliency map and laplacian equator bias fit for the sitting condition.

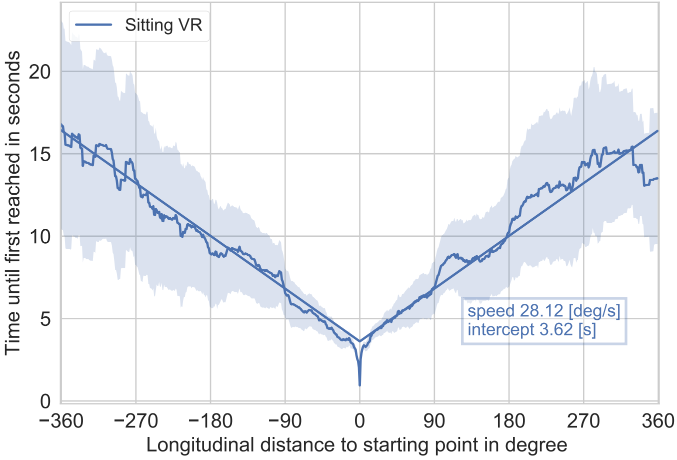


Figure C.3: Exploration time computed as the average time until a specific longitudinal offset from the starting point is reached, for the sitting condition.

C.2.4 *Does scene content affect viewing behavior?*

C.2.4.1 *Scenes used for the analysis*

In Figure C.4 we show the eight scenes with lower (top) and higher (bottom) entropy analyzed in the main text.

C.2.4.2 *Viewer's behavior metrics*

Measuring viewer's behavior in an objective manner is not a simple task. First, we define *salient regions* as the 5% most salient pixels of a scene as described in the main text. We then rely on three metrics recently proposed by Serrano et al. [292] in the context of gaze analysis for VR movie editing (time to reach a salient region (*timeToSR*), percentage of fixations inside the salient regions (*percFixInside*), and number of fixations (*nFix*)), and propose a fourth, novel one, tailored for static 360° panoramas:

TIME TO REACH A SALIENT REGION (*timetosr*) It accounts for the time before the user fixates on a salient region.

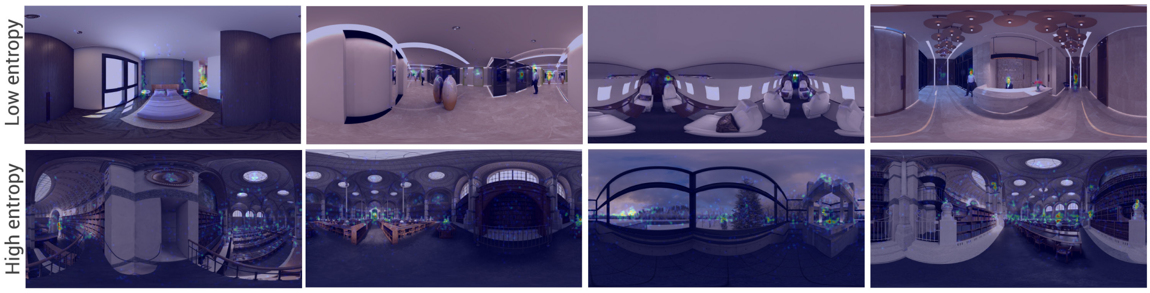


Figure C.4: Scenes with lower (top) and higher (bottom) entropy in our dataset.

Percentage of fixations inside the salient regions ($percfixinside$) It is computed after the user fixates on a salient region for the first time, in order to make it independent of *timeToSR*. It is indicative of the interest of the user in the salient regions.

Number of fixations ($nfix$) Computed as the ratio between the number of fixations and the number of gaze samples.

Convergence time ($convergtime$) For every scene, we obtain the per-user saliency maps at different time steps, and compute the similarity (CC score) with the fully-converged saliency map. We plot the temporal evolution of this CC score, and compute the area under this curve. This metric represents the temporal convergence of saliency maps; it is inversely proportional to how long it takes for the fixation map during exploration to converge to the ground truth saliency map.

C.2.4.3 *Analysis*

VR condition The design of our experiment implies that the same user can see several (but not all) of the tested conditions, so we first need to test for independence of observations (Wald's test). For one of the metrics (*percFixInside*) the effect of the user was found to be not significant ($p = 0.071$), while for the other three (*timeToSR* and *nFix*, and *convergTime*) it was significant ($p = 0.023$, and $p < 0.001$, and $p < 0.001$, respectively). We therefore during the main text employ ANOVA when analyzing *percFixInside*, since the samples are considered to be independent, and report significance values obtained from multilevel modeling [39, 262] (which is well-suited for related data) for the other three metrics. Results of this analysis are shown in Table C.1, Table C.2, Table C.3, and Table C.4.

An important question, is whether the particular point (viewport) at which the viewer starts exploring the scene has an influence when exploring the panorama. To make this analysis tractable, we have tested four equally-spaced *viewports*[26] per scene, covering the 360°. We define two possible values for each viewport, $\{V_0, V_1\}$, corresponding to whether or not they contain salient regions.

We find that the *viewport* condition has a significant influence on *timeToSR*, which is not surprising ($p < 0.001$): If it does not contain salient regions (V_0), viewers take longer to reach them elsewhere. There is also a significant influence on *convergTime* ($p = 0.004$); specifically, users converge faster towards the final saliency map when the initial viewport contains salient regions (V_1).

There is no influence of the viewport on the number of fixations, nor in the number of fixations inside salient regions. We also evaluate whether the starting viewport conditions the final saliency map for a given scene: For each scene, we compute the similarity between the final saliency map of the i^{th} viewport and the other three, using again the CC score. We obtain a median CC score

[26] Viewports match the FOV of the Oculus, 97°. Thus the four viewports define slightly overlapping areas.

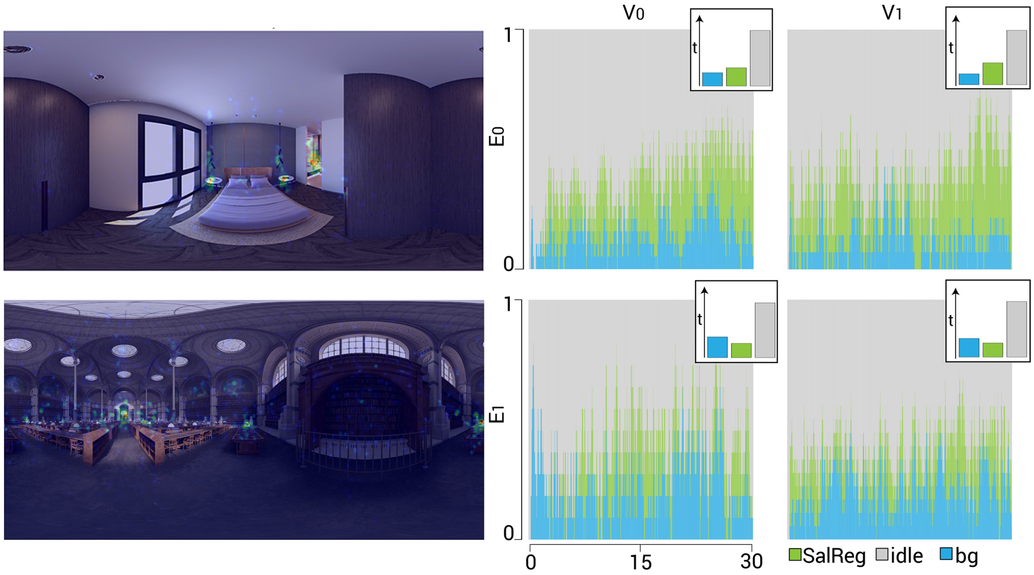


Figure C.5: State sequence distribution for two representative scenes. Insets show the time spent in each of the states. Users spend more time in salient regions for scenes with low entropy (top), and their attention gets attracted faster than for scenes with high entropy (bottom). The time spent until users fixate on salient regions also depends on their starting viewport, being this time shorter when their initial field of view contains salient regions (right) than when there is no regions of interest (left).

of 0.79, which indicates that the final saliency maps after 30 seconds, starting from different viewports, converge and are very similar.

We have visually confirmed the findings presented in the main text by performing a *state sequences* analysis [106, 292]. For each temporal step we classify users' fixations in one of three different states, corresponding to whether they fixate on a salient region, the background, or there is no fixation (idle state). This last state corresponds to saccadic movements taking place. This analysis allows us to observe the temporal evolution of the gaze pattern, as well as the time spent in each of the states. Results for two representative scenes are shown in Figure C.5. As already confirmed by our metrics, it shows how the time spent until fixating on a salient regions depends on both the entropy and the viewport, and that the number of fixations inside salient regions is highly dependent on the distribution of salient regions (entropy) of the scene.

Table C.1: Tests of fixed effects (indicating whether each predictor has a significant influence) for timeToSR for the *VR* condition.

	viewport	entropy	viewport*entropy
F	17.206	31.599	2.235
$(df_{\text{num}}, df_{\text{den}})$	(1, 288.908)	(1, 298.270)	(1, 289.746)
p	0.000	0.000	0.136

DESKTOP CONDITION For the *desktop* condition for all our four metrics the effect of the user was found to be not significant. We therefore employ ANOVA when analyzing all of them. Results of this analysis are shown in Table C.5, Table C.6, Table C.7, and Table C.8.

Table C.2: Tests of fixed effects (indicating whether each predictor has a significant influence) for nFix for the *VR* condition.

	viewport	entropy	viewport*entropy
F	0.072	5.296	0.053
$(df_{\text{num}}, df_{\text{den}})$	(1, 227.323)	(1, 232.769)	(1, 222.793)
p	0.789	0.022	0.818

Table C.3: Tests of fixed effects (indicating whether each predictor has a significant influence) for convergTime for the *VR* condition.

	viewport	entropy	viewport*entropy
F	8.259	38.785	2.527
$(df_{\text{num}}, df_{\text{den}})$	(1, 259.394)	(1, 269.978)	(1, 255.295)
p	0.004	0.000	0.113

C.2.5 *What else do we learn from head and gaze statistics?*

In this section we present additional details on the number and duration of fixations in Table C.9. We also show in Figure C.6 the vestibulo-ocular reflex for both *VR* conditions, and in Figures C.7, C.8, C.9, and C.10 the head speed and gaze offset distributions for both *VR* conditions.

Table C.4: ANOVA results for percFixInside for the *VR* condition.

	viewport	entropy	viewport*entropy
F	3.338	208.832	0.110
(df_1, df_2)	(1, 306)	(1, 306)	(1, 306)
p	0.069	0.000	0.740

Table C.5: ANOVA results for timeToSR for the *desktop* condition.

	viewport	entropy	viewport*entropy
F	29.083	73.159	20.225
(df_1, df_2)	(1, 306)	(1, 306)	(1, 306)
p	0.000	0.000	0.000

Table C.6: ANOVA results for nFix for the *desktop* condition.

	viewport	entropy	viewport*entropy
F	2.345	7.428	1.067
(df_1, df_2)	(1, 306)	(1, 306)	(1, 306)
p	0.127	0.007	0.302

Table C.7: ANOVA results for convergTime for the *desktop* condition.

	viewport	entropy	viewport*entropy
F	6.743	152.707	0.210
(df_1, df_2)	(1, 306)	(1, 306)	(1, 306)
p	0.010	0.000	0.647

Table C.8: ANOVA results for percFixInside for the *desktop* condition.

	viewport	entropy	viewport*entropy
F	9.377	336.481	3.089
(df_1, df_2)	(1, 306)	(1, 306)	(1, 306)
p	0.002	0.000	0.080

Table C.9: Duration and number of fixations for the three conditions.

	Fixation duration	Number of fixations
VR	267 ± 121	50.35 ± 14.63
Seated	243 ± 107	42.64 ± 15.24
Desktop	270 ± 138	55.03 ± 12.42

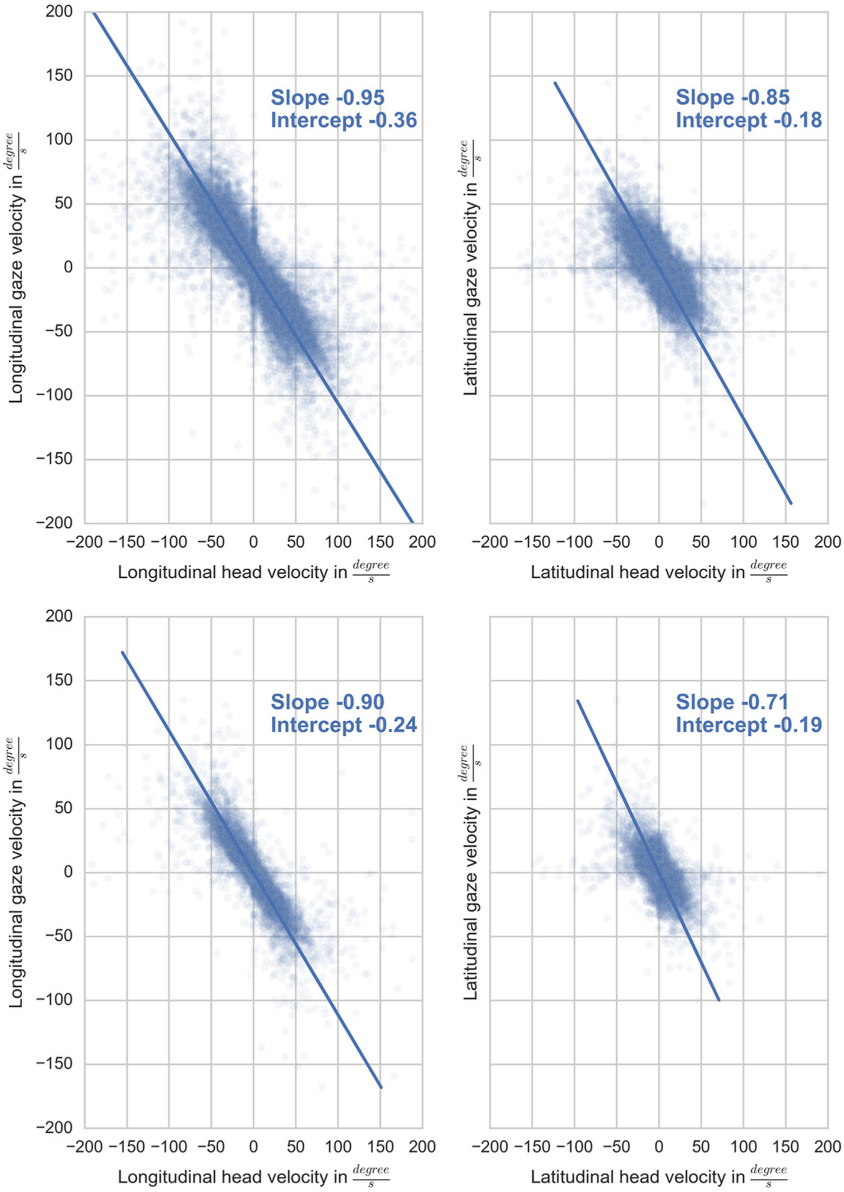
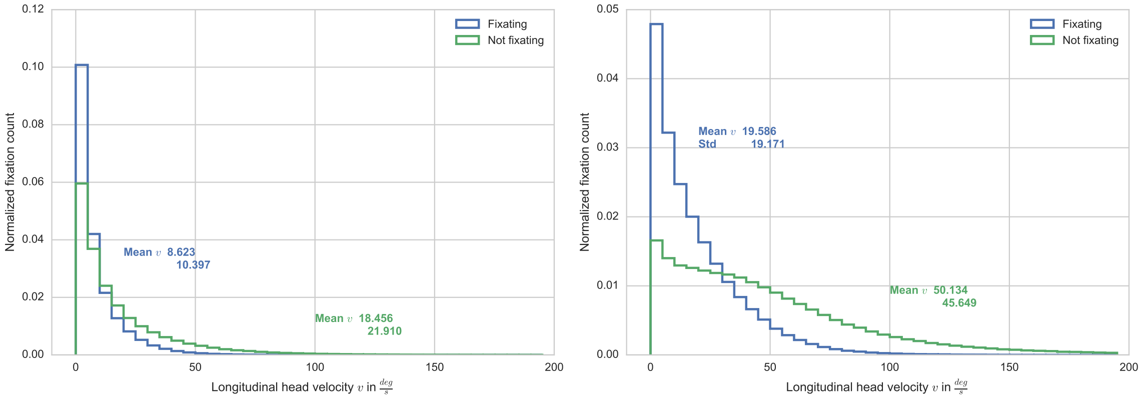


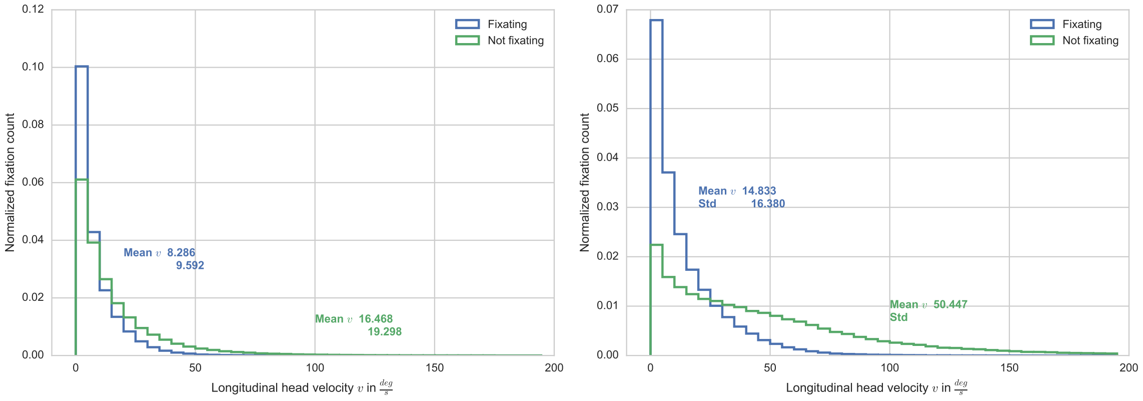
Figure C.6: The vestibulo-ocular reflex can be identified in the subset of points where fixations were detected. Top row: *VR* condition. Bottom row: *Seated* condition. Left corresponds to longitudinal velocity, and right to the latitudinal velocity.



(a)

(b)

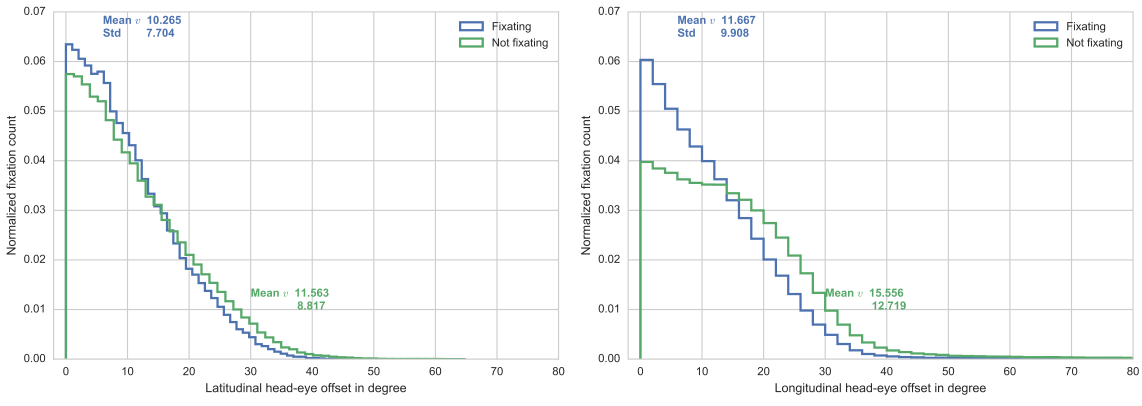
Figure C.7: Latitudinal and longitudinal head speed distributions when fixating or not for the *VR* condition. When fixating, the longitudinal head speed is significantly lower - this allows to detect when users are likely fixating, even without the use of an eye tracker. Though the latitudinal head speed is also slightly lower, the effect is mainly visible in the longitudinal component of head velocity.



(a)

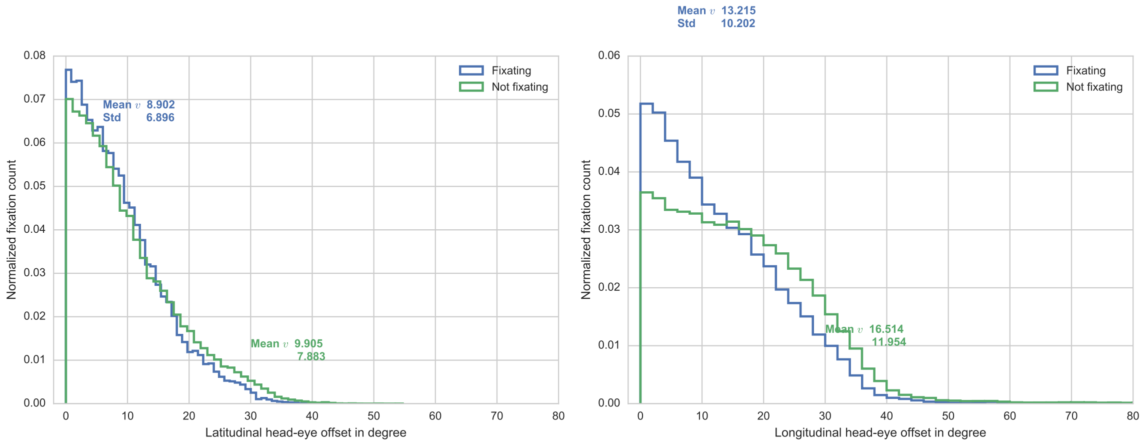
(b)

Figure C.8: Latitudinal and longitudinal head speed distributions when fixating or not for the *seated* condition. When fixating, the longitudinal head speed is significantly lower - this allows to detect when users are likely fixating, even without the use of an eye tracker. Though the latitudinal head speed is also slightly lower, the effect is mainly visible in the longitudinal component of head velocity.



(a) (b)

Figure C.9: Latitudinal and longitudinal eye eccentricities (offset between head and eye orientation in degrees) when fixating or not for the *VR* condition. When fixating, the longitudinal eye eccentricity is significantly smaller than when not fixating. The same does not seem to be true for the latitudinal eye eccentricity, however. Combined with the insights from Figure C.7, this is a strong prior for gaze location when the head is moving slowly.



(a) (b)

Figure C.10: Latitudinal and longitudinal eye eccentricities (offset between head and eye orientation in degrees) when fixating or not for the *seated* condition. When fixating, the longitudinal eye eccentricity is significantly smaller than when not fixating. The same does not seem to be true for the latitudinal eye eccentricity, however. Combined with the insights from Figure C.7, this is a strong prior for gaze location when the head is moving slowly.

D. Movie editing and cognitive event segmentation in virtual reality video: additional details and analysis

D. 1 Stimuli generation

In Table D.2 we include the complete set of conditions tested during our experiments, and in Table D.1 we indicate the different cutting techniques used for each type of edit.

E1	Jump cut
E2	Compressed time
	Match-on-action

Table D.1: Cutting techniques used for each type of edit.

D. 2 Analysis details

D. 2.1 *Statistical analysis*

As explained in the main text, we use factorial ANOVA and multilevel modeling to analyze the metrics. We analyze main effects and first order interactions. In Tables D.3 to D.6 we report ANOVA and Bonferroni post hoc results for *framesToROI* and *scanpathError*, two of our four metrics. For the metrics *nFix* and *percFixInside* we carry out multilevel regression with the subject as a random effect and the four variables of interest and their first-order interactions as fixed effects. We report here, in Tables D.8 to D.13, the tests of fixed effects, which indicate whether the corresponding predictor (variable of interest) has a significant influence on the outcome (the value of the metric). For the regression, as per standard practice, we used dummy variables to code the categorical variables with more than two values that we had (A, $\mathrm{R_a}$ and $\mathrm{R_b}$); thus, each variable with n levels is recoded into $n-1$ variables: R_b is recoded into $\mathrm{R_{b,1}}$ and $\mathrm{R_{b,2}}$, as shown in Table D.7, and R_a and A are recoded in an analogous manner.

A_0	$R_{b,0}$	$R_{a,0}$	E_1	$(A_0, R_{b,0}, R_{a,0}, E_1)$
			E_2	$(A_0, R_{b,0}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_0, R_{b,0}, R_{a,1}, E_1)$
			E_2	$(A_0, R_{b,0}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_0, R_{b,0}, R_{a,2}, E_1)$
			E_2	$(A_0, R_{b,0}, R_{a,2}, E_2)$
	$R_{b,1}$	$R_{a,0}$	E_1	$(A_0, R_{b,1}, R_{a,0}, E_1)$
			E_2	$(A_0, R_{b,1}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_0, R_{b,1}, R_{a,1}, E_1)$
			E_2	$(A_0, R_{b,1}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_0, R_{b,1}, R_{a,2}, E_1)$
			E_2	$(A_0, R_{b,1}, R_{a,2}, E_2)$
	$R_{b,2}$	$R_{a,0}$	E_1	$(A_0, R_{b,2}, R_{a,0}, E_1)$
			E_2	$(A_0, R_{b,2}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_0, R_{b,2}, R_{a,1}, E_1)$
			E_2	$(A_0, R_{b,2}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_0, R_{b,2}, R_{a,2}, E_1)$
			E_2	$(A_0, R_{b,2}, R_{a,2}, E_2)$
A_{40}	$R_{b,0}$	$R_{a,0}$	E_1	$(A_{40}, R_{b,0}, R_{a,0}, E_1)$
			E_2	$(A_{40}, R_{b,0}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_{40}, R_{b,0}, R_{a,1}, E_1)$
			E_2	$(A_{40}, R_{b,0}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_{40}, R_{b,0}, R_{a,2}, E_1)$
			E_2	$(A_{40}, R_{b,0}, R_{a,2}, E_2)$
	$R_{b,1}$	$R_{a,0}$	E_1	$(A_{40}, R_{b,1}, R_{a,0}, E_1)$
			E_2	$(A_{40}, R_{b,1}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_{40}, R_{b,1}, R_{a,1}, E_1)$
			E_2	$(A_{40}, R_{b,1}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_{40}, R_{b,1}, R_{a,2}, E_1)$
			E_2	$(A_{40}, R_{b,1}, R_{a,2}, E_2)$
	$R_{b,2}$	$R_{a,0}$	E_1	$(A_{40}, R_{b,2}, R_{a,0}, E_1)$
			E_2	$(A_{40}, R_{b,2}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_{40}, R_{b,2}, R_{a,1}, E_1)$
			E_2	$(A_{40}, R_{b,2}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_{40}, R_{b,2}, R_{a,2}, E_1)$
			E_2	$(A_{40}, R_{b,2}, R_{a,2}, E_2)$
A_{80}	$R_{b,0}$	$R_{a,0}$	E_1	$(A_{80}, R_{b,0}, R_{a,0}, E_1)$
			E_2	$(A_{80}, R_{b,0}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_{80}, R_{b,0}, R_{a,1}, E_1)$
			E_2	$(A_{80}, R_{b,0}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_{80}, R_{b,0}, R_{a,2}, E_1)$
			E_2	$(A_{80}, R_{b,0}, R_{a,2}, E_2)$
	$R_{b,1}$	$R_{a,0}$	E_1	$(A_{80}, R_{b,1}, R_{a,0}, E_1)$
			E_2	$(A_{80}, R_{b,1}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_{80}, R_{b,1}, R_{a,1}, E_1)$
			E_2	$(A_{80}, R_{b,1}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_{80}, R_{b,1}, R_{a,2}, E_1)$
			E_2	$(A_{80}, R_{b,1}, R_{a,2}, E_2)$
	$R_{b,2}$	$R_{a,0}$	E_1	$(A_{80}, R_{b,2}, R_{a,0}, E_1)$
			E_2	$(A_{80}, R_{b,2}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_{80}, R_{b,2}, R_{a,1}, E_1)$
			E_2	$(A_{80}, R_{b,2}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_{80}, R_{b,2}, R_{a,2}, E_1)$
			E_2	$(A_{80}, R_{b,2}, R_{a,2}, E_2)$

Table D.2: Complete set of conditions tested during our experiments.

Table D.3: ANOVA results for *scanpathError*.

	A	R_a	R_b	E	$A * R_a$	$A * R_b$	A*E	$R_a * R_b$	$R_a * E$	$R_b * E$
F	14.511	168.569	1.660	0.038	1.116	7.567	9.032	0.793	6.356	6.503
(df_1, df_2)	(2, 787)	(2, 787)	(2, 787)	(1, 787)	(4, 787)	(4, 787)	(2, 787)	(4, 787)	(2, 787)	(2, 787)
p	0.000	0.000	0.191	0.846	0.348	0.000	0.000	0.530	0.002	0.002

Table D.4: ANOVA results for *framesToROI*.

	A	R_a	R_b	E	$A * R_a$	$A * R_b$	A*E	$R_a * R_b$	$R_a * E$	$R_b * E$
F	198.059	10.300	6.478	1.373	3.998	7.096	0.949	2.601	1.688	0.379
(df_1, df_2)	(2, 787)	(2, 787)	(2, 787)	(1, 787)	(4, 787)	(4, 787)	(2, 787)	(4, 787)	(2, 787)	(2, 787)
p	0.000	0.000	0.002	0.242	0.003	0.000	0.388	0.035	0.185	0.685

Table D.5: Significance of pairwise comparisons computed with a Bonferroni post hoc for *scanpathError*.

	A_0	A_{40}	A_{80}
A_0	-	0.277	0.000
A_{40}	0.277	-	0.001
A_{80}	0.000	0.001	-

	$R_{b,0}$	$R_{b,1}$	$R_{b,2}$
$R_{b,0}$	-	1.000	0.216
$R_{b,1}$	1.000	-	0.458
$R_{b,2}$	0.216	0.458	-

	$R_{a,0}$	$R_{a,1}$	$R_{a,2}$
$R_{a,0}$	-	0.804	0.000
$R_{a,1}$	0.804	-	0.000
$R_{a,2}$	0.000	0.000	-

Table D.6: Significance of pairwise comparisons computed with a Bonferroni post hoc for *framesToROI*.

	A_0	A_{40}	A_{80}
A_0	-	0.000	0.000
A_{40}	0.000	-	0.000
A_{80}	0.000	0.000	-

	$R_{b,0}$	$R_{b,1}$	$R_{b,2}$
$R_{b,0}$	-	0.056	1.000
$R_{b,1}$	0.056	-	0.004
$R_{b,2}$	1.000	0.004	-

	$R_{a,0}$	$R_{a,1}$	$R_{a,2}$
$R_{a,0}$	-	0.078	0.000
$R_{a,1}$	0.078	-	0.112
$R_{a,2}$	0.000	0.112	-

Table D.7: Recoding of the categorical variable R_b (which has three levels, 0-2) into two dummy variables $R_{b,1}$ and $R_{b,2}$.

R_b	$R_{b,1}$	$R_{b,2}$
0	0	0
1	1	0
2	0	1

Table D.8: Tests of fixed effects (indicating whether each predictor has a significant influence) for *nFix* (I).

	A_{80}	A_{40}	$R_{b,2}$	$R_{b,1}$	$R_{a,2}$	$R_{a,1}$	E	$A_{80} * R_{b,2}$
F	8.873	0.003	0.901	0.015	2.577	0.639	0.622	0.102
$(df_{\text{num}}, df_{\text{den}})$	(1, 825.026)	(1, 823.819)	(1, 820.260)	(1, 819.617)	(1, 822.773)	(1, 820.867)	(1, 824.704)	(1, 824.045)
p	0.003	0.954	0.343	0.902	0.109	0.424	0.430	0.749

Table D.9: Tests of fixed effects (indicating whether each predictor has a significant influence) for *nFix* (II).

	$A_{80} * Rb,1$	$A_{80} * R_{a,2}$	$A_{80} * R_{a,1}$	$A_{80} * E$	$A_{40} * R_{b,2}$	$A_{40} * Rb,1$	$A_{40} * R_{a,2}$	$A_{40} * R_{a,1}$
F	0.580	2.968	1.706	0.162	0.084	0.145	0.020	0.030
$(df_{\text{num}}, df_{\text{den}})$	(1, 827.664)	(1, 818.866)	(1, 823.559)	(1, 819.236)	(1, 825.343)	(1, 824.988)	(1, 822.832)	(1, 826.654)
p	0.447	0.085	0.192	0.687	0.772	0.704	0.888	0.862

Table D.10: Tests of fixed effects (indicating whether each predictor has a significant influence) for *nFix* (III).

	$A_{40} * E$	$R_{b,2} * R_{a,2}$	$R_{b,2} * R_{a,1}$	$R_{b,2} * E$	$R_{b.1} * R_{a,2}$	$R_{b,1} * R_{a,1}$	$R_{b,1} * E$	$R_{a,2} * E$	$R_{a,1} * E$
F	0.845	0.344	0.021	1.538	0.337	1.262	0.194	0.350	2.277
$(df_{\text{num}}, df_{\text{den}})$	(1, 824.431)	(1, 815.187)	(1, 815.496)	(1, 814.542)	(1, 814.828)	(1, 816.124)	(1, 813.199)	(1, 819.715)	(1, 816.461)
p	0.358	0.558	0.884	0.215	0.562	0.262	0.660	0.554	0.132

Table D.11: Tests of fixed effects (indicating whether each predictor has a significant influence) for *percFixInside* (I).

	A_{80}	A_{40}	$R_{b,2}$	$R_{b,1}$	$R_{a,2}$	$R_{a,1}$	E	$A_{80} * R_{b,2}$
F	12.687	2.206	0.070	5.946	0.050	4.633	0.177	0.025
$(df_{\text{num}}, df_{\text{den}})$	(1, 826.742)	(1, 825.687)	(1, 822.587)	(1, 822.000)	(1, 824.795)	(1, 823.086)	(1, 826.481)	(1, 825.847)
p	0.000	0.138	0.792	0.015	0,824	0.032	0.674	0.874

Table D.12: Tests of fixed effects (indicating whether each predictor has a significant influence) for *percFixInside* (II).

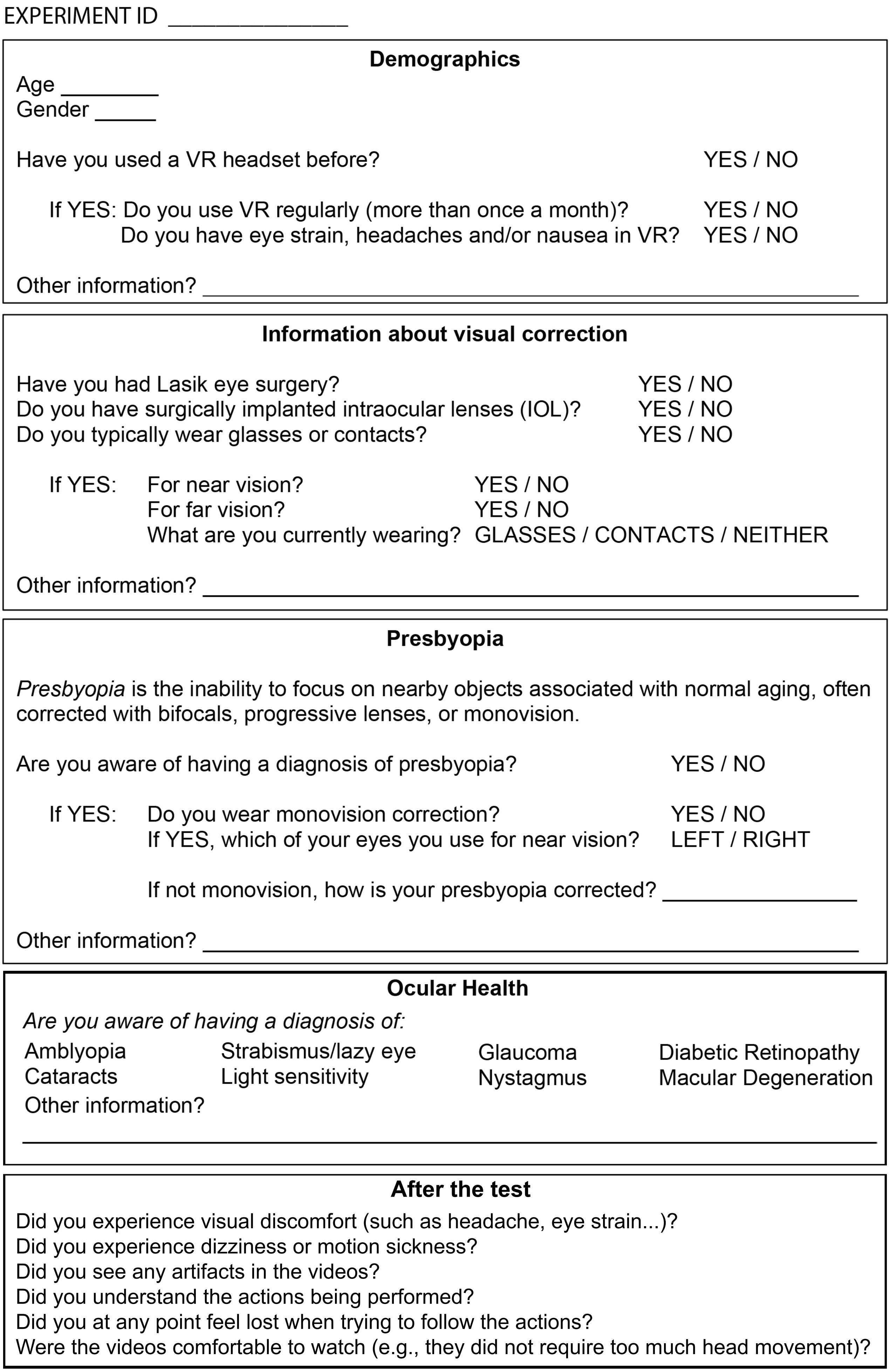
	$A_{80} * Rb,1$	$A_{80} * R_{a,2}$	$A_{80} * R_{a,1}$	$A_{80} * E$	$A_{40} * R_{b,2}$	$A_{40} * Rb,1$	$A_{40} * R_{a,2}$	$A_{40} * R_{a,1}$
F	0.119	4.681	5.992	4.022	0.654	2.583	0.525	3.154
$(df_{\text{num}}, df_{\text{den}})$	(1, 828.965)	(1, 821.317)	(1, 825.431)	(1, 821.607)	(1, 826.987)	(1, 826.654)	(1, 824.760)	(1, 828.107)
p	0.731	0.031	0.015	0.045	0.419	0.108	0.469	0.076

Table D.13: Tests of fixed effects (indicating whether each predictor has a significant influence) for *percFixInside* (III).

	$A_{40} * E$	$R_{b,2} * R_{a,2}$	$R_{b,2} * R_{a,1}$	$R_{b,2} * E$	$R_{b,1} * R_{a,2}$	$R_{b,1} * R_{a,1}$	$R_{b,1} * E$	$R_{a,2} * E$	$R_{a,1} * E$
F	1.131	0.186	0.328	0.024	0.000	0.536	0.374	3.852	2.081
$(df_{\text{num}}, df_{\text{den}})$	(1, 826.198)	(1, 818.114)	(1, 818.372)	(1, 817.539)	(1, 817.775)	(1, 818.878)	(1, 816.329)	(1, 822.044)	(1, 819.169)
p	0.288	0.666	0.567	0.878	0.993	0.464	0.541	0.050	0.150

D. 3 Questionnaire

In this section we include the questionnaire that users were asked to fill before performing the test. It includes information about demographics, visual correction (if any), and general ocular health. The last part of the questionnaire includes the questions we asked to the users after the test during the debriefing session.



EXPERIMENT ID ________________

Demographics

Age _________
Gender _____

Have you used a VR headset before? YES / NO

If YES: Do you use VR regularly (more than once a month)? YES / NO
Do you have eye strain, headaches and/or nausea in VR? YES / NO

Other information? ____

Information about visual correction

Have you had Lasik eye surgery? YES / NO
Do you have surgically implanted intraocular lenses (IOL)? YES / NO
Do you typically wear glasses or contacts? YES / NO

If YES: For near vision? YES / NO
For far vision? YES / NO
What are you currently wearing? GLASSES / CONTACTS / NEITHER

Other information? ____

Presbyopia

Presbyopia is the inability to focus on nearby objects associated with normal aging, often corrected with bifocals, progressive lenses, or monovision.

Are you aware of having a diagnosis of presbyopia? YES / NO

If YES: Do you wear monovision correction? YES / NO
If YES, which of your eyes you use for near vision? LEFT / RIGHT

If not monovision, how is your presbyopia corrected? ________________

Other information? ____

Ocular Health

Are you aware of having a diagnosis of:

Amblyopia	Strabismus/lazy eye	Glaucoma	Diabetic Retinopathy
Cataracts	Light sensitivity	Nystagmus	Macular Degeneration

Other information?

____

After the test

Did you experience visual discomfort (such as headache, eye strain...)?
Did you experience dizziness or motion sickness?
Did you see any artifacts in the videos?
Did you understand the actions being performed?
Did you at any point feel lost when trying to follow the actions?
Were the videos comfortable to watch (e.g., they did not require too much head movement)?

E. Motion parallax for 360 RGBD video: additional details

E. 1 Depth improvement for motion parallax

We show more analyses of influence of the parameters in Equation 8.4 in Chapter 8, and the edge stopping function used for the edge guidance term (Equation 8.6 in Chapter 8).

E.1.1 *Parameter Analysis*

We show the influence of each term in Equation 8.4 in in Chapter 8 (Figure E.1). The first two rows show the results from different parameter sets of the data term (Equation 8.5). As the data parameter λ_{data} increases, the result gets similar to the input depth map that has artifacts. On the other hand, if λ_{data} is too small, our method overly corrects the input depth, resulting in wrong depth. The smoothness term (Equation 8.7) enforces smoothness of depth values, so the higher the parameter λ_{sm} is, the smoother the result becomes. However, if λ_{sm} is not enough large, our method fails to correct the regions in which wrong depth inputs should be smoothed out (See the region that is indicated by a white arrow in the case of $\lambda_{\text{sm}} = 5 \cdot 10^{-3}$). The edge guidance term takes edge-aware propagation into account. If λ_e is too small, bleeding/blurry artifacts cannot be removed well, and if it is too large, it overly corrects the input depth, resulting in wrong depth.

E.1.2 *Edge-stopping Function*

The edge guidance term (Equation 8.6 in Chapter 8) includes a spatially-varying weight $w_e(i,j)$ consisting of an edge-stopping function: Tukey's biweight. The edge-stopping function prevents propagation across edges, so the choice of it is crucial in our problem. The colorization method of Levin et al. [188] uses an exponential function to give edge-awareness:

$$f_{\exp}(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right), \tag{E.1}$$

where σ is a 3×3 local standard deviation. The anisotropic diffusion method of Perona and Malick [255], they employ Lorentzian edge-stopping function:

$$f_{\text{lorentzian}}(x) = \frac{1}{1 + \frac{x^2}{2\sigma^2}}. \tag{E.2}$$

Black et al. [29] suggest using Tukey's biweight as an edge-aware function:

$$f_{\text{tukey}}(x) = \begin{cases} \left(1 - \frac{x^2}{\sigma^2}\right)^2 & |x| \le \sigma \\ 0 & \textit{otherwise} \end{cases}. \tag{E.3}$$

For all the functions in depth improvement, x is a local gradient value in 3×3 neighborhoods, which is denoted as $e(i) - e(j)$ where e is either an RGB or a depth pixel, i is an index of a pixel of interest, and j is i's local neighborhood.

Figure E.2 shows plots of the three functions with $\sigma^2 = 0.5$. All the plots show that around zero they have value close to one, and they rapidly drop in the other way around. In depth cleaning, x-axis is a gradient value of a guidance image, so we can see that if an edge-stopping function has a small value, it is likely that there is a prominent gradient (edge), and vice versa. Lorentzian function shows the most generous behavior in determining edges since it has the thickest tails.

The exponential function plummets as it goes away from zero, but Tukey's biweight used in our method shows the strictest determinant of gradients. In other words, if there is even a small gradient, Tukey's biweight blocks propagation during optimization while Lorentzian allowing. The double arrows in Figure E.2 indicate the intervals that the functions allow propagation. Figure E.3 shows the results from different edge-stopping functions. Lorentzian function shows the smoothest result since its gradient determinant is generous. The exponential function gives better results, but around non-prominent edges, it shows blurry depth boundaries. Tukey's biweight produces the sharpest outputs since its gradient threshold is the strictest.

E. 2 layered video representation

E.2.1 *Computation of opacity map α: Intermediate steps*

We show a more detailed pipeline of our α processing explained in Section 8.3 in Chapter 8. See Figure E.4 for the description of the pipeline.

E.2.2 *Computation of maximum parallax ρ*

In this section we include additional details on how to compute the maximum parallax angle ρ, this dictates the number of pixels to inpaint in the *inpainted* background layer. The geometrical representation for the following derivation is given in Figure E.5. We provide the derivation for the 2D case for simplicity, since the extension for 3D space is straight-forward. Given two consecutive mesh vertices d_1 and d_2, the maximum parallax ρ depends on the depth difference between such vertices, and the maximum head movement allowed (r_{max}). The vectors v_2 and v_3 are given by the position of the two vertices, (d_1^x, d_1^y) and (d_2^x, d_2^y). The vector v_1 is determined by setting a maximum head displacement circle around the center of projection (x_0, y_0) with radius r_{max}. The angle of maximum parallax ρ can be expressed as:

$$\begin{aligned}
&\cos(\rho) = \tfrac{v_4 \cdot v_5}{\|v_4\|\|v_5\|},\\
&\textit{with}\\
&v_5 = v_3 - v_1,\\
&v_4 = v_2 - v_1
\end{aligned} \tag{E.4}$$

We aim to compute the maximum ρ, so we can formulate our problem as a maximization problem in the form:

$$\underset{\mathrm{x_1,y_1 \leq r}}{\arg\max}\, \rho = \arccos \tfrac{v_4 \cdot v_5}{\|v_4\|\|v_5\|} \tag{E.5}$$

If we substitute v_4 and v_5 by their expressions as defined in Equation E.4:

$$\underset{\mathrm{x_1,y_1 \leq r}}{\arg\max} \arccos \frac{(d_{\mathrm{x}}^2 - x_1)(d_{\mathrm{x}}^1 - x_1) + (d_{\mathrm{y}}^2 - y_1)(d_{\mathrm{y}}^1 - y_1)}{\sqrt{(d_{\mathrm{x}}^2 - x_1)^2 + (d_{\mathrm{y}}^2 - y_1)^2}\sqrt{(d_{\mathrm{x}}^1 - x_1)^2 + (d_{\mathrm{y}}^1 - y_1)^2}} \tag{E.6}$$

Finally, we can assume that maximum angle of parallax ρ will lie in the outer boundary, such that $x_1^2 + y_1^2 = r_{\text{max}}^2$:

$$\underset{\mathrm{|x_1| \leq r_{max}}}{\arg\max} \arccos \frac{(d_{\mathrm{x}}^2 - x_1)(d_{\mathrm{x}}^1 - x_1) + (d_{\mathrm{y}}^2 - \sqrt{r^2 - x_1^2})(d_{\mathrm{y}}^1 - \sqrt{r^2 - x_1^2})}{\sqrt{(d_{\mathrm{x}}^2 - x_1)^2 + (d_{\mathrm{y}}^2 - \sqrt{r^2 - x_1^2})^2}\sqrt{(d_{\mathrm{x}}^1 - x_1)^2 + (d_{\mathrm{y}}^1 - \sqrt{r^2 - x_1^2})^2}} \tag{E.7}$$

where all the terms but x_1 are known, and we seek to find x_1 such that $|x_1| \leq r_{\text{max}}$. By knowing x_1 and y_1 we can go back to Equation E.5 and obtain the maximum angle of parallax ρ. By knowing ρ, we can compute P (assuming a planar surface to inpaint, i.e., for this example this will imply that P lies at the same distance than the further mesh vertex d_2 in the y axis) to calculate the number of pixels N_{pix} to inpaint in the *inpainted* layer.

E. 3 Results

We show additional examples of results as described in Section 8.6 in Chapter 8. Six results are shown in Figure E.6, depicting a variety of scenes. For each result, we show a view from the center of projection, and a displaced view leading to disocclusions. In each of the scenes we highlight regions illustrating the added parallax. We also include results from monocular videos from different sources with estimated depth in Figure E.7.

E. 4 Evaluation

We show additional plots from the experiments explained in Section 8.5 in Chapter 8, and the questionnaires used for user study.

E.4.1 *Experiment #1*

We conduct a user study without noticing participants about the difference between two visualizations modes: with/without head-motion parallax (Experiment #1 in Chapter 8). Figure E.8 shows representative frames of the dataset used for this user study.

E.4.2 *Experiment #2*

We conduct a user study to check sickness in VR depending on whether motion parallax is supported or not (Experiment #2 in Chapter 8). Figure E.9 shows representative frames of the dataset used for this user study.

E.4.3 *Experiment #3*

We conduct a user study with noticing participants about the difference between two visualizations modes: with/without head-motion parallax (Experiment #3 in Chapter 8). Figure E.10 shows representative frames of the dataset used for this user study.

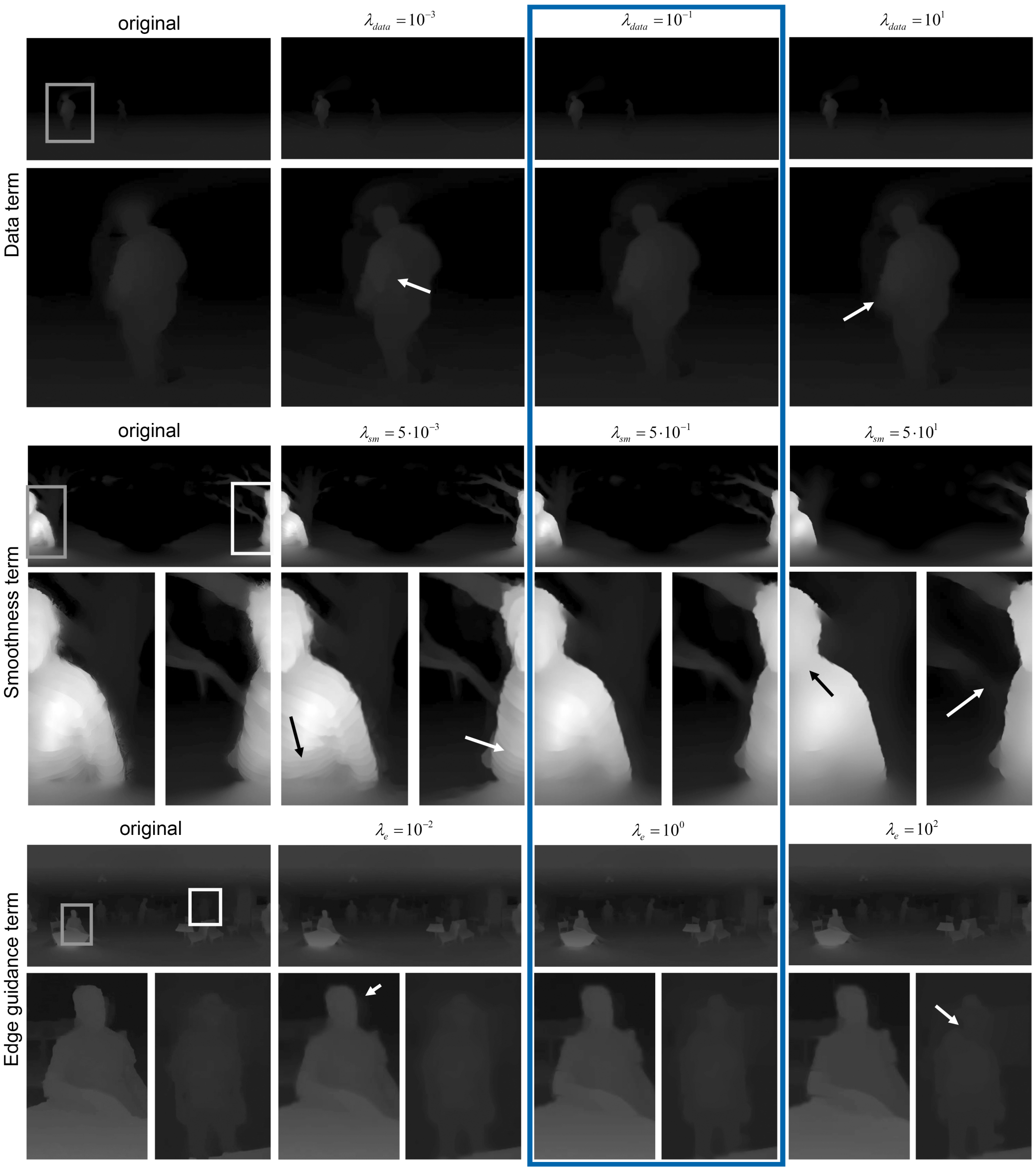


Figure E.1: Results with different parameters. The third column is our parameter choice (marked with a blue square).

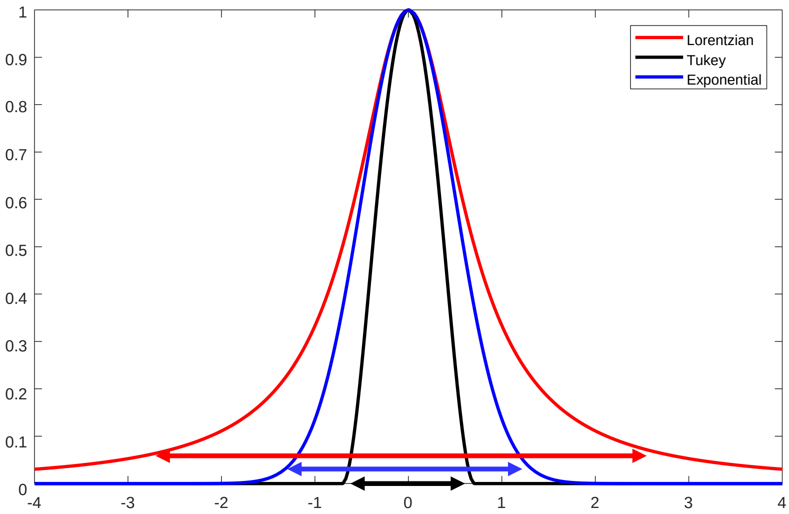


Figure E.2: Edge stopping functions. For our results we choose the Tukey stopping function.

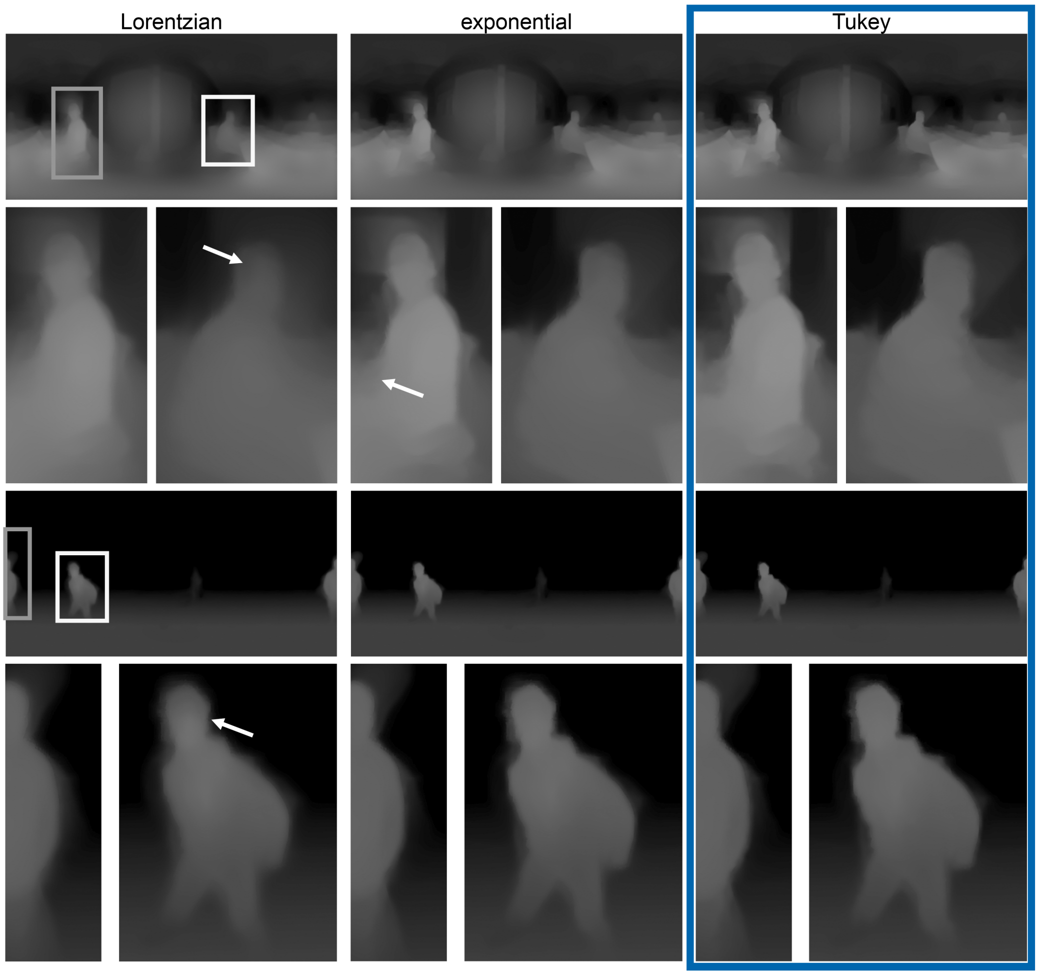
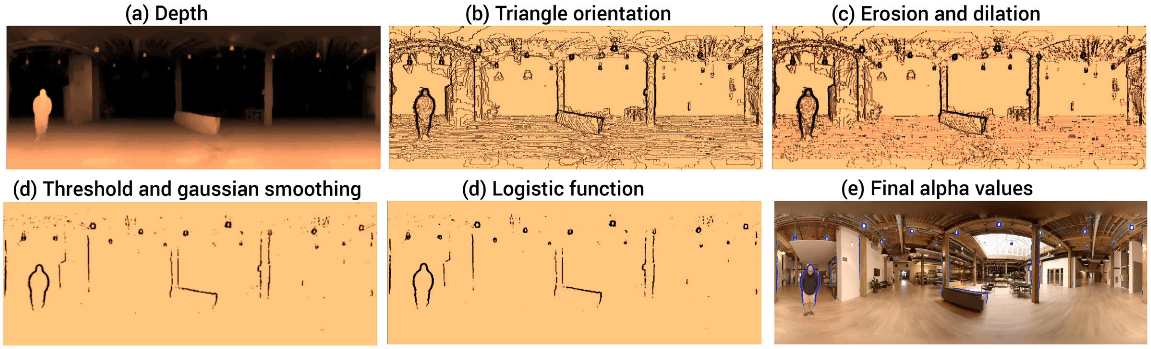


Figure E.3: Results from different edge-stopping functions. For our results we choose the Tukey stopping function (marked with a blue square).


(a) Depth

(b) Triangle orientation

(c) Erosion and dilation

(d) Threshold and gaussian smoothing

(d) Logistic function

(e) Final alpha values

Figure E.4: (a) Depth map of the foreground layer for a given frame of the video. (b) Raw fragment ori-entation map O^F. (c) Intermediate orientation map processed with erosion and dilation operators $(O^F \bullet K)$. (d) Processed fragment orientation map $\hat{\alpha}^F$. The raw orientation values $(O^F_{i,j})$ are too noisy to provide smooth transparency values, as they include information not only on disocclusions, but also on foreshortening and quantization errors. We process this raw values as described in Chapter 8, and obtain a clean transparency map $(\hat{\alpha}^F)$ that smoothly matches the RGB image. (e) RGB image of which the processed fragment orientation map $\hat{\alpha}^F$ is overlaid on top.

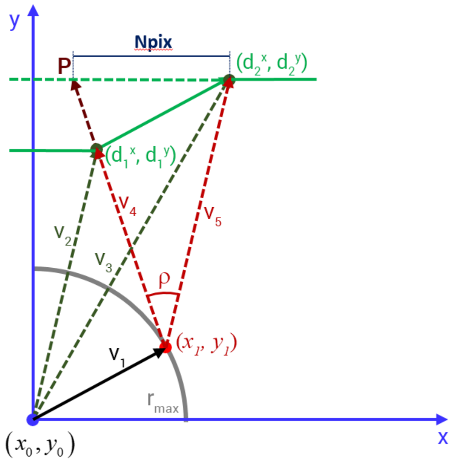


Figure E.5: Illustration that accompanies the computation of maximum parallax ρ (see main text in this section).

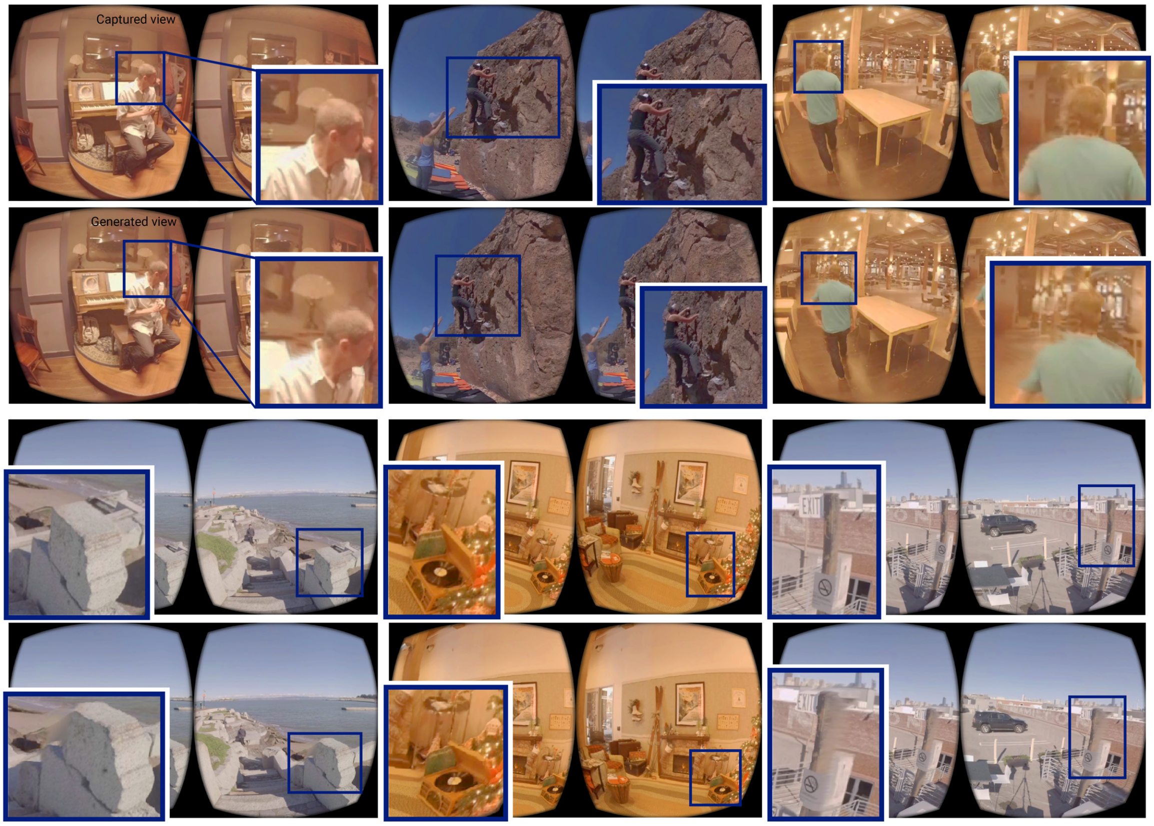


Figure E.6: Six examples of novel views generated with our method. For each example we show the original view (top), and the corresponding displaced view (bottom). We also include close-ups of regions where the added parallax is clearly visible.

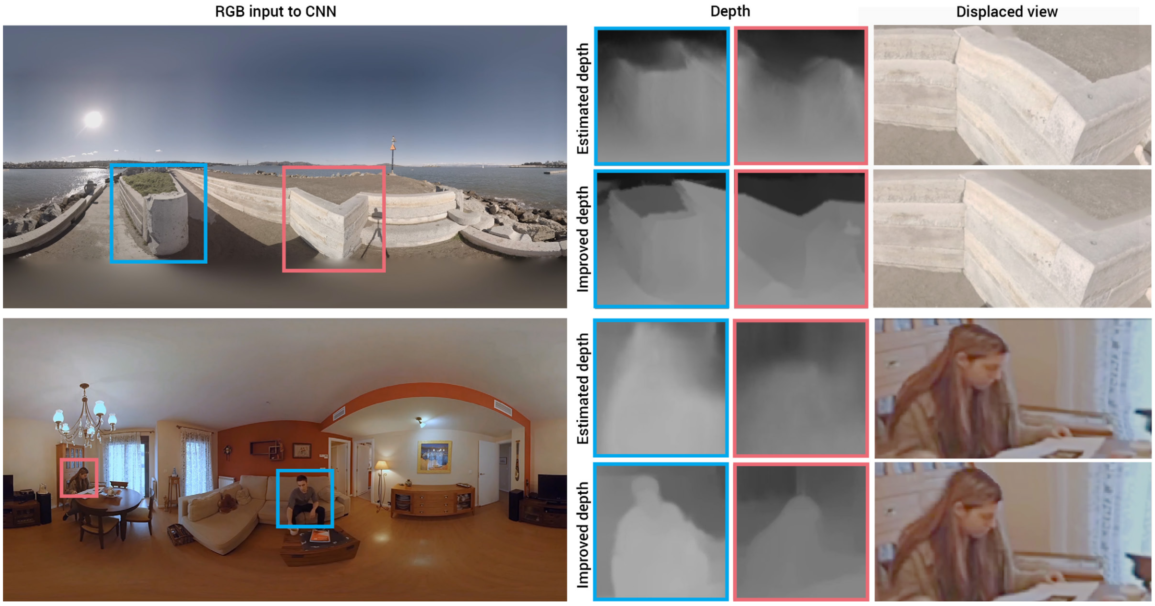


Figure E.7: Results using monocular video without depth as input. For each row, we show a representative frame (left), details of the initial estimated depth [112] and our improved depth (middle), as well as the corresponding results when generating a novel view (right). Our method yields minimal distortions even in the presence of such suboptimal input. The first row depicts a scene captured for this work with a GoPro Odyssey, while the second row is taken from a previously existing short clip (dataset from [292]); in both cases we use only a 360° RGB monocular view as input and drop the remaining data.

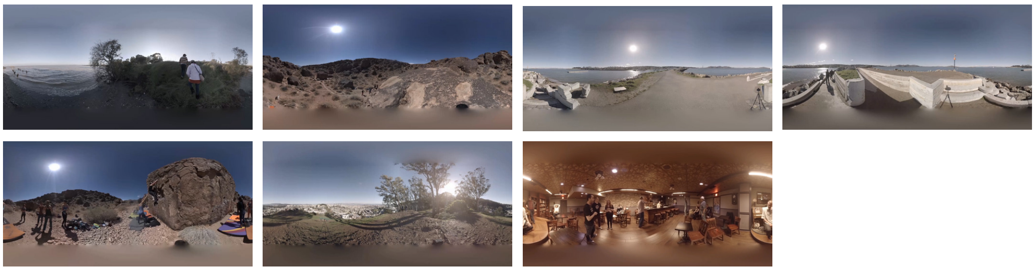


Figure E.8: Representative frames of Experiment #1.

Figure E.9: Representative frames of Experiment #2.

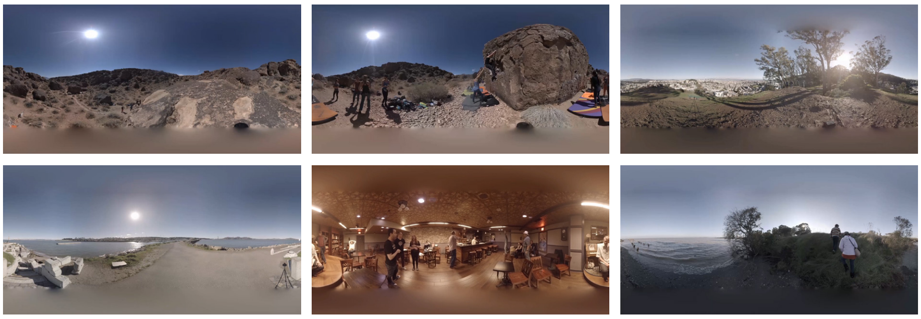


Figure E.10: Representative frames of Experiment #3 .

E.4.4 *Questionnaires*

We include the questionnaires used for the user study described in Section 8.5 in Chapter 8. The first questionnaire is to ask participants if they are familiar with a VR headset, prior to the experiments. The second questionnaire is used to ask preference of one viewing mode over the other in terms of realism, comfort, immersion, presence of visual artifacts, and global preference. To assess if the subjects experienced sickness, dizziness, and/or vertigo, and whether they had experienced discomfort, and if so, to measure how much they felt those during viewing, we use the sickness questionnaire.

EXPERIMENT ID ______________________

Age __________

Gender _______

Have you used an HMD before? YES / NO

If yes:

Was it PC-based or smartphone-based?

How many times have you used it? _____________________________

When was the last time you used it?____________________________

Do you use it regularly (more than once a month)? YES / NO

Have you ever experienced eyestrain, dizziness, headaches, or nausea in VR? YES / NO

Do you play videogames regularly? YES/NO

If yes, how often (days/week)?____

Other information?____

Experiment #1 and Experiment #3

EXPERIMENT ID _______________
RENDERING MODE _______________
VIDEO ___________________

1. With which of the two methods did you feel more immersed in the scene?

First method ☐ Second method ☐

Comments:...
....

2. Which of the two methods had more visual artifacts?

First method ☐ Second method ☐

Comments:...
....

3. Which of the two methods offers a more realistic experience?

First method ☐ Second method ☐

Comments:...
....

4. With which of the two methods did you feel more comfortable watching the scene?

First method ☐ Second method ☐

Comments:...
....

5. Globally, which of the two methods do you prefer for visualizing the scene?

First method ☐ Second method ☐

Comments:...
....

6. [This is a simple question regarding the scene content, it differs for each scene]

....

SICKNESS QUESTIONNAIRE (Experiment #2)

EXPERIMENT ID ________________

RENDERING MODE ________________

General discomfort	○ None	○ Slight	○ Moderate	○ Severe
Fatigue	○ None	○ Slight	○ Moderate	○ Severe
Eyestrain	○ None	○ Slight	○ Moderate	○ Severe
Difficulty focusing	○ None	○ Slight	○ Moderate	○ Severe
Headache	○ None	○ Slight	○ Moderate	○ Severe
Fullness of head	○ None	○ Slight	○ Moderate	○ Severe
Blurred vision	○ None	○ Slight	○ Moderate	○ Severe
Dizzy (eyes closed)	○ None	○ Slight	○ Moderate	○ Severe
Vertigo	○ None	○ Slight	○ Moderate	○ Severe

Did you feel totally comfortable while watching the videos?

____

Did you experience at any moment while watching the videos any symptom of dizziness, vertigo, disorientation, or nausea? If yes, describe the video and situation where this happened.

____

Bibliography

[1] E.H. Adelson. "On seeing stuff: the perception of materials by humans and machines." In: *Proc. of Photonics West 2001-electronic imaging*. International Society for Optics and Photonics. 2001, pp. 1–12.

[2] E.H. Adelson and J.R. Bergen. "The plenoptic function and the elements of early vision." In: *Computational models of visual processing* (1991).

[3] C. Aguerrebere, A. Almansa, J. Delon, Y. Gousseau, and P. Muse. "Single Shot High Dynamic Range Imaging Using Piecewise Linear Estimators." In: *Proc. of IEEE International Conference on Computational Photography (ICCP)*. 2014.

[4] M. Aharon, M. Elad, and A. Bruckstein. "K-SVD: An algorithm for Designing Overcomplete Dictionaries for Sparse Representation." In: *IEEE Trans. on Signal Processing* 54 (2006), pp. 4311–4322.

[5] M. Aittala, T. Weyrich, and J. Lehtinen. "Two-shot SVBRDF capture for stationary materials." In: *ACM Trans. Graph.* 34.4 (2015), 110:1–13.

[6] O. Akyüz and E. Reinhard. "Color appearance in high dynamic range imaging." In: *SPIE Journal of Electronic Imaging* 15.3 (2006).

[7] C. Aliaga, C. Castillo, D. Gutierrez, M.A. Otaduy, J. Lopez-Moreno, and A. Jarabo. "An Appearance Model for Textile Fibers." In: *Computer Graphics Forum* 36.4 (2017).

[8] M. Allue, A. Serrano, M.G. Bedia, and B. Masia. "Crossmodal Perception in Immersive Environments." In: *Proc. of CEIG*. Ed. by Alejandro Garcia-Alonso and B. Masia. 2016.

[9] M.S.C. Almeida and M.A.T. Figueiredo. "Frame-based Image Deblurring with Unknown Boundary Conditions using the Alternating Direction Method of Multipliers." In: *Proc. of IEEE International Conference on Image Processing (ICIP)*. 2013, pp. 582–585.

[10] X. An, X. Tong, J.D. Denning, and F. Pellacini. "AppWarp: Retargeting Measured Materials by Appearance-space Warping." In: *ACM Trans. Graph.* 30.6 (2011), 147:1–147:10.

[11] J.D. Anderson. *The Reality of Illusion: An Ecological Approach to Cognitive Film Theory*. Southern Illinois University Press, 1996.

[12] R. Anderson, D. Gallup, J.T. Barron, et al. "Jump: Virtual Reality Video." In: *ACM Trans. Graph.* 35.6 (2016), 198:1–198:13.

[13] R. Anderson, D. Gallup, J.T. Barron, J. Kontkanen, N. Snavely, C. Hernández, S. Agarwal, and S.M. Seit. "Jump: Virtual Reality Video." In: *ACM Trans. Graph.* (2016).

[14] I. Arev, H. Park, Y. Sheikh, J.K. Hodgins, and A. Shamir. "Automatic editing of footage from multiple social cameras." In: *ACM Trans. Graph.* 33.4 (2014), 81:1–81:11.

[15] M. Ashikhmin, S. Premože, and P. Shirley. "A Microfacet-based BRDF Generator." In: *Proc. of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '00. 2000, pp. 65–74.

[16] A. Atapour-Abarghouei and T.P. Breckon. "A comparative review of plausible hole filling strategies in the context of scene depth image completion." In: *Computers & Graphics* 72 (2018), pp. 39–58.

[17] S.H. Baek, D. Gutierrez, and M.H. Kim. "Birefractive Stereo Imaging for Single-Shot Depth Acquisition." In: *ACM Trans. Graph.* 35.6 (2016).

[18] S.H. Baek, I. Kim, D. Gutierrez, and M.H. Kim. "Compact Single-Shot Hyperspectral Imaging Using a Prism." In: *ACM Trans. Graph.* 36.6 (2017), 217:1–12.

[19] D. Baricevic, T. Höllerer, and M. Turk. “Densification of Semi-Dense Reconstructions for Novel View Generation of Live Scenes.” In: *Proc. of IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2017, pp. 842–851.

[20] J. T. Barron, A. Adams, Y. Shih, and C. Hernández. “Fast bilateral-space stereo for synthetic defocus.” In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 4466–4474.

[21] M. Batz, T. Ritcher, J.U. Garbas, A. Papts, J. Seiler, and A. Kaup. “High Dynamic Range Video Reconstruction From a Stereo Camera Setup.” In: *Elsevier* 29.2 (2014).

[22] S. Bell, P. Upchurch, N. Snavely, and K. Bala. “OpenSurfaces: a richly annotated catalog of surface appearance.” In: *ACM Trans. Graph.* 32.4 (2013), 111:1–111:17.

[23] A. Ben-Artzi, R. Overbeck, and R. Ramamoorthi. “Real-time BRDF Editing in Complex Lighting.” In: *ACM Trans. Graph.* 25.3 (2006), pp. 945–954.

[24] A. Ben-Artzi, K. Egan, F. Durand, and R. Ramamoorthi. “A Precomputed Polynomial Representation for Interactive BRDF Editing with Global Illumination.” In: *ACM Trans. Graph.* 27.2 (2008), 13:1–13:13.

[25] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester. “Image inpainting.” In: *Proc. of SIGGRAPH*. ACM, 2000, pp. 417–424.

[26] F. Berthouzoz, W. Li, and M. Agrawala. “Tools for placing cuts and transitions in interview video.” In: *ACM Trans. Graph.* 31.4 (2012), 67:1–67:8.

[27] I. Biederman. “Recognition-by-Components: A Theory of Human Image Understanding.” In: *Psychological Review* 94 (1987), pp. 115–147.

[28] S. Birchfield and C. Tomasi. “Depth Discontinuities by Pixel-to-Pixel Stereo.” In: *International Journal of Computer Vision* 35.3 (1999), pp. 269–293.

[29] M.J. Black, G. Sapiro, D.H. Marimont, and D. Heeger. “Robust anisotropic diffusion.” In: *IEEE Trans. Image Processing* 7.3 (1998), pp. 421–432.

[30] P. Blignaut. “Fixation identification: The optimum threshold for a dispersion algorithm.” In: *Attention, Perception, & Psychoph.* 71.4 (2009), pp. 881–895.

[31] G. Boccignone and M. Ferraro. “Modelling gaze shift as a constrained random walk.” In: *Physica A: Statistical Mechanics and its Applications* 331.1 (2004), pp. 207–218.

[32] D. Bordwell, K. Thompson, and J. Ashton. *Film art: An introduction*. Vol. 7. McGraw-Hill New York, 1997.

[33] A. Borji and L. Itti. “State-of-the-Art in Visual Attention Modeling.” In: *IEEE Trans. on PAMI* 35.1 (2013), pp. 185–207.

[34] A. Bousseau, J.P. O’shea, F. Durand, R. Ramamoorthi, and M. Agrawala. “Gloss Perception in Painterly and Cartoon Rendering.” In: *ACM Trans. Graph.* 32.2 (2013), 18:1–18:13.

[35] I. Boyadzhiev, K. Bala, S. Paris, and E. Adelson. “Band-Sifting Decomposition for Image-Based Material Editing.” In: *ACM Trans. Graph.* 34.5 (2015), p. 163.

[36] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers.” In: *Foundations and Trends in Machine Learning* 1.3 (2011), pp. 127–239.

[37] H. Bristow, A. Eriksson, and S. Lucey. “Fast Convolutional Sparse Coding.” In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 391–398.

[38] H. Bristow and S. Lucey. “Optimization Methods for Convolutional Sparse Coding.” In: *arXiv:1406.2407*. 2014.

[39] W. Browne and J. Rasbash. “Multilevel Modelling.” In: *Handbook of data analysis*. Hardy, M. and Bryman, A. (Eds.) Sage Publications, 2004, pp. 459–478.

[40] C. Buehler, M. Bosse, L. McMillan, S.J. Gortler, and M.F. Cohen. “Unstructured lumigraph rendering.” In: *Proc. of SIGGRAPH*. ACM, 2001, pp. 425–432.

[41] B. Burley. “Physically based shading at Disney.” In: *ACM SIGGRAPH Courses*. 2012.

[42] P. Buyssens, M. Daisy, D. Tschumperlé, and O. Lézoray. “Depth-aware patch-based image disocclusion for virtual view synthesis.” In: *SIGGRAPH Asia Technical Briefs*. ACM, 2015, 2:1–2:4.

[43] P. Buyssens, O. Le Meur, M. Daisy, D. Tschumperlé, and O. Lézoray. “Depth-Guided Disocclusion Inpainting of Synthesized RGB-D Images.” In: *IEEE Trans. Image Processing* 26.2 (2017), pp. 525–538.

[44] Z. Bylinskii, P. Isola, C. Bainbridge, A. Torralba, and A. Oliva. “Intrinsic and extrinsic effects on image memorability.” In: *Vision research* 116 (2015), pp. 165–178.

[45] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand. “What do different evaluation metrics tell us about saliency models?” In: *arXiv:1604.03605* (2016).

[46] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba. *MIT Saliency Benchmark*. http://saliency.mit.edu. Last accessed: 2018-03-05. 2017.

[47] E. Candes, J. Romberg, and T. Tao. “Robust incertainty principles: exact signal reconstruction from highly incomplete frequency information.” In: *IEEE Trans. on Information Theory* 52.2 (2006), pp. 489–509.

[48] E. Candes, Y. Eldarb, D. Needella, and P. Randallc. “Compressed sensing with coherent and redundant dictionaries.” In: *Applied and Computational Harmonic Analysis* 31.1 (2010).

[49] J.M. Carroll and T.G. Bever. “Segmentation in cinema perception.” In: *Science* 191.4231 (1976), pp. 1053–1055.

[50] S. Castillo, T. Judd, and D. Gutierrez. “Using Eye-Tracking to Assess Different Image Retargeting Methods.” In: *Proc. of Symposium on Applied Perception in Graphics and Visualization (APGV)*. Toulouse, France: ACM Press, 2011.

[51] S. Chaabouni, J. Benois-Pineau, O. Hadar, and C.B. Amar. “Deep Learning for Saliency Prediction in Natural Video.” In: *arXiv preprint:1604.08010* (2016).

[52] G. Chandler. *Cut by cut*. Michael Wiese Productions, 2004.

[53] T. Chang, Y. Chou, and J. Yang. “Robust depth enhancement and optimization based on advanced multilateral filters.” In: *EURASIP J. Adv. Sig. Proc.* 2017 (2017), p. 51.

[54] S. Chaudhuri, E. Kalogerakis, S. Giguere, and T. Funkhouser. “AttribIt: Content Creation with Semantic Attributes.” In: *Proc. of ACM Symposium on User Interface Software and Technology (UIST)*. ACM, 2013.

[55] G. Chaurasia, O. Sorkine, and G. Drettakis. “Silhouette-Aware Warping for Image-Based Rendering.” In: *Comput. Graph. Forum* 30.4 (2011), pp. 1223–1232.

[56] G. Chaurasia, S. Duchêne, O. Sorkine-Hornung, and G. Drettakis. “Depth synthesis and local warps for plausible image-based navigation.” In: *ACM Trans. Graph.* 32.3 (2013), 30:1–30:12.

[57] B. Chen, G. Polatkan, G. Sapiro, D. Blei, D. Dunson, and L. Carin. “Deep Learning with Hierarchical Convolutional Factor Analysis.” In: *IEEE Trans. on PAMI* 35.8 (2013), pp. 1887–1901.

[58] J. Chen, S. Paris, and F. Durand. “Real-time Edge-aware Image Processing with the Bilateral Grid.” In: *ACM Trans. Graph.* 26.3 (2007). ISSN: 0730-0301.

[59] M.M. Cheng, N.J. Mitra, X. Huang, P.H.S. Torr, and S.M. Hu. “Global contrast based salient region detection.” In: *IEEE Trans. on PAMI* 37.3 (2015), pp. 569–582.

[60] E. Cheslack-Postava, R. Wang, O. Akerlund, and F. Pellacini. "Fast, Realistic Lighting and Material Design Using Nonlinear Cut Approximation." In: *ACM Trans. Graph.* 27.5 (2008), 128:1–128:10.

[61] H. Cho, S.J. Kim, and S. Lee. "Single-shot High Dynamic Range Imaging Using Coded Electronic Shutter." In: *Computer Graphics Forum* (2014).

[62] I. Choi, D.S. Jeon, G. Nam, D. Gutierrez, and M.H. Kim. "High-Quality Hyperspectral Reconstruction Using a Spectral Prior." In: *ACM Trans. Graph.* 36.6 (2017), 218:1–13.

[63] R. Choudhury. *Principles of Colour and Appearance Measurement*. Woodhead Publishing, 2014.

[64] D.B. Christianson, S.E. Anderson, L. He, D.H. Salesin, D.S. Weld, and M.F. Cohen. "Declarative camera control for automatic cinematography." In: *Proc. of the thirteenth national conference on Artificial intelligence, Vol. 1*. 1996, pp. 148–155.

[65] M. Christie, R. Machap, J.M. Normand, P. Olivier, and J.H. Pickering. "Virtual Camera Planning: A Survey." In: *Proc. of Int. Symposium on Smart Graphics*. 2005, pp. 40–52.

[66] N. Cohn. "Visual Narrative Structure." In: *Cognitive Science* 37.3 (2013), pp. 413–452.

[67] M. Colbert and S. Pattanaik. "BRDF-Shop: Creating Physically Correct Bidirectional Reflectance Distribution Functions." In: *IEEE Computer Graphics and Applications* (2006), pp. 30–36.

[68] R. Cong, J. Lei, C. Zhang, Q. Huang, X. Cao, and C. Hou. "Saliency Detection for Stereoscopic Images Based on Depth Confidence Analysis and Multiple Cues Fusion." In: *IEEE Signal Processing Letters* 23.6 (2016), pp. 819–823.

[69] Cornell. *Reflectance Database - Cornell University Program of Computer Graphics*. http://www.graphics.cornell.edu/online/measurements/reflectance/index.html. Last accessed: 2018-03-05. 2001.

[70] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. "A Deep Multi-Level Network for Saliency Prediction." In: *Proc. of International Conference on Pattern Recognition (ICPR)*. 2016.

[71] A. Criminisi, P. Pérez, and K. Toyama. "Region filling and object removal by exemplar-based image inpainting." In: *IEEE Trans. Image Processing* 13.9 (2004), pp. 1200–1212.

[72] J. Cutting. "Perceiving Scene in Film and in the World." In: *Moving image theory: ecological considerations*. Ed. by J. D. Anderson and B. F. Anderson. 2004. Chap. 1, pp. 9–26.

[73] J.E. Cutting. "Event segmentation and seven types of narrative discontinuity in popular movies." In: *Acta psychologica* 149 (2014), pp. 69–77.

[74] F. Danieau, A. Guillo, and R. Doré. "Attention guidance for immersive video content in head-mounted displays." In: *Proc. of IEEE Virtual Reality*. 2017, pp. 205–206.

[75] N.M. Davis, A. Zook, B. O'Neill, B. Headrick, M. Riedl, A. Grosz, and M. Nitsche. "Creativity support for novice digital filmmaking." In: *Proc. of ACM SIGCHI*. 2013, pp. 651–660.

[76] P. Debevec. *Light Probe Image Gallery*. http://www.pauldebevec.com/Probes/. Last accessed: 2017-05-05. 1998.

[77] P.E. Debevec, G. Borshukov, and Y. Yu. "Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping." In: *Proc. of the 9th Eurographics Rendering Workshop*. 1998.

[78] P.E. Debevec and J. Malik. "Recovering High Dynamic Range Radiance Maps from Photographs." In: *Proc. of ACM SIGGRAPH*. 1997, pp. 369–378.

[79] P.E. Debevec, C. J. Taylor, and J. Malik. "Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach." In: *Proc. of SIGGRAPH*. ACM, 1996, pp. 11–20.

[80] F. Devernay and A.R. Peon. “Novel View Synthesis for Stereoscopic Cinema: Detecting and Removing Artifacts.” In: *Proc. of the 1st International Workshop on 3D Video Processing*. 3DVP ’10. Firenze, Italy: ACM, 2010, pp. 25–30. ISBN: 978-1-4503-0159-6.

[81] S. DiVerdi, A. Hertzmann, and A. Serrano. *VR Parallax Rendering*. Patent. Pending number: P7437-US.

[82] P. Didyk, P. Sitthi-Amorn, W. Freeman, F. Durand, and W. Matusik. “Joint view expansion and filtering for automultiscopic 3D displays.” In: *ACM Trans. Graph.* 32.6 (2013), p. 221.

[83] E. Dmytryk. *On Film Editing. An Introduction to thè Art of Film Construction*. 1984.

[84] P. Dollár and C.L. Zitnick. “Structured Forests for Fast Edge Detection.” In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2013, pp. 1841–1848.

[85] Y. Dong, J. Wang, F. Pellacini, X. Tong, and B. Guo. “Fabricating spatially-varying subsurface scattering.” In: *ACM Trans. Graph.* 29.4 (2010), p. 62.

[86] D.L. Donoho. “Compressed sensing.” In: *IEEE Trans. on Information Theory* 52.4 (2006), pp. 1289–1306.

[87] J. Dorsey, H. Rushmeier, and F.X. Sillion. *Digital Modeling of Material Appearance*. Morgan Kaufmann, 2007.

[88] A. Doshi and M.M. Trivedi. “Head and eye gaze dynamics during visual attention shifts in complex environments.” In: *Journal of Vision* 12.2 (2012), p. 9.

[89] S. Du, B. Masia, S. Hu, and D. Gutierrez. “A Metric of Visual Comfort for Stereoscopic Motion.” In: *ACM Trans. Graph.* 32.6 (2013), 222:1–9.

[90] A.T. Duchowski, N. Cournia, and H. Murphy. “Gaze-contingent displays: a review.” In: *Cyberpsychol Behav.* 7.6 (2004), pp. 621–34.

[91] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. “Least angle regression.” In: *Annals of statistics* 32 (2004), pp. 407–499.

[92] M. Eisemann, B. de Decker, M.A. Magnor, P. Bekaert, E. de Aguiar, N. Ahmed, C. Theobalt, and A. Sellent. “Floating Textures.” In: *Comput. Graph. Forum* 27.2 (2008), pp. 409–418.

[93] A. El Gamal. *High dynamic range image sensors*. Proc. of International Solid-State Circuits Conference Tutorials. 2002.

[94] M. Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. 1st. Springer Publishing Company, Inc., 2010.

[95] Unreal Engine. *Virtual Reality Best Practices*. https://docs.unrealengine.com/en-us/Platforms/VR/ContentSetup. Last accessed: 2018-09-14. 2017.

[96] S. Ershov, K. Kolchin, and K. Myszkowski. “Rendering Pearlescent Appearance Based On Paint-Composition Modelling.” In: *Computer Graphics Forum* 20.3 (2001).

[97] C. Eugene. *Measurement of total visual appearance: a CIE challenge of soft metrology*. Tech. rep. 12th IMEKO TC1-TC7 joint Symposium on Man, Science & Measurement, 2008.

[98] R.S. Feris, R. Raskar, L. Chen, K. Tan, and M. Turk. “Discontinuity Preserving Stereo with Small Baseline Multi-Flash Illumination.” In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2005, pp. 412–419.

[99] J. Filip and R. Vávra. “Template-Based Sampling of Anisotropic BRDFs.” In: *Computer Graphics Forum* (2014).

[100] R. A. Fisher. “On the interpretation of chi square from contingency tables, and the calculation of P.” In: *Journal of the Royal Statistical Society* 85 (1922), pp. 87–94.

[101] R. W. Fleming, C. Wiebel, and K. Gegenfurtner. “Perceptual qualities and material classes.” In: *Journal of Vision* 13.8 (2013), pp. 9–9.

[102] R.W. Fleming, R.O. Dror, and E.H. Adelson. "Real-world illumination and the perception of surface reflectance properties." In: *Journal of Vision* 3 (2003), pp. 347–368.

[103] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. "DeepStereo: Learning to Predict New Views From the World's Imagery." In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[104] A. Fores, J. Ferwerda, J. Gu, and X. Zhao. "Toward a Perceptually based Metric for BRDF Modeling." In: *Proc. of 20th Color and Imaging Conference*. CIC'12. 2012, pp. 142–148.

[105] E.G. Freedman. "Coordination of the eyes and head during visual orienting." In: *Experimental brain research* 190.4 (2008), pp. 369–387.

[106] A. Gabadinho, G. Ritschard, N. Müller, and M. Studer. "Analyzing and Visualizing State Sequences in R with TraMineR." In: *Journal of Statistical Software* 40.1 (2011).

[107] O. Gallo, N. Gelfand, W. Chen, M. Tico, and K. Pulli. "Artifact-free High Dynamic Range Imaging." In: 2009.

[108] Q. Galvane, R. Ronfard, C. Lino, and M. Christie. "Continuity editing for 3d animation." In: *Proc. of AAAI Conference on Artificial Intelligence*. 2015.

[109] E. Garces, A. Agarwala, D. Gutierrez, and A. Hertzmann. "A Similarity Measure for Illustration Style." In: *ACM Trans. Graph.* 33.4 (2014).

[110] A. Gibaldi, M. Vanegas, P.J. Bex, and G. Maiello. "Evaluation of the Tobii EyeX Eye tracking controller and Matlab toolkit for research." In: *Behav. Res. Methods* (2016).

[111] I. Gkioulekas, B. Xiao, S. Zhao, E.H. Adelson, T. Zickler, and K. Bala. "Understanding the role of phase function in translucent appearance." In: *ACM Trans. Graph.* 32.5 (2013), 147:1–147:19.

[112] C. Godard, O. Mac Aodha, and G.J. Brostow. "Unsupervised Monocular Depth Estimation with Left-Right Consistency." In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2017, pp. 6602–6611.

[113] M. Goesele, J. Ackermann, S. Fuhrmann, C. Haubold, R. Klowsky, D. Steedly, and R. Szeliski. "Ambient point clouds for view interpolation." In: *ACM Trans. Graph.* 29.4 (2010), 95:1–95:6.

[114] S. Goferman, L. Zelnik-Manor, and A. Tal. "Context-aware saliency detection." In: *IEEE Trans. PAMI* 34.10 (2012), pp. 1915–26.

[115] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. "The Lumigraph." In: *Proc. of SIGGRAPH*. ACM, 1996, pp. 43–54.

[116] M. Granados, K.I. Kim, J. Tompkin, and C. Theobalt. "Automatic Noise Modeling for Ghost-free HDR Reconstruction." In: *ACM Trans. Graph.* 32.6 (2013), 201:1–201:10.

[117] S. Grogorick, M. Stengel, E. Eisemann, and M. Magnor. "Subtle Gaze Guidance for Immersive Environments." In: *Proc. of the ACM Symposium on Applied Perception (SAP)*. SAP '17. Cottbus, Germany, 2017, 4:1–4:7.

[118] R.B. Grosse, R. Raina, H. Kwong, and A.Y. Ng. "Shift-invariant sparse coding for audio classification." In: *Proc. of Conference on Uncertainty in Artificial Intelligence (UAI)*. 2007, pp. 149–158.

[119] Y. Gryaditskaya, T. Pouli, E. Reinhard, K. Myszkowski, and H.-P. Seidel. "Motion aware exposure bracketing for HDR video." In: *Computer Graphics Forum*. Vol. 34. 4. 2015, pp. 119–130.

[120] J. Gu, Y. Hitomi, T. Mitsunaga, and S.K. Nayar. "Coded Rolling Shutter Photography: Flexible Space-Time Sampling." In: *Proc. of IEEE International Conference on Computational Photography (ICCP)*. 2010.

[121] D. Guarnera, G.C. Guarnera, A. Ghosh, C. Denk, and M. Glencross. "BRDF Representation and Acquisition." In: *Computer Graphics Forum* 35 (2016), pp. 625–650.

[122] F. Guo, J. Shen, and X. Li. "Learning to detect stereo saliency." In: *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*. 2014, pp. 1–6.

[123] A. Gupta, P. Bhat, M. Dontcheva, B. Curless, O. Deussen, and M. Cohen. "Enhancing and experiencing spacetime resolution with videos and stills." In: *Proc. of IEEE International Conference on Computational Photography (ICCP)*. 2009.

[124] D. Gutierrez, B. Masia, and A. Jarabo. "Computational Photography." In: *Proc. of CEIG (Tutorials)*. 2012.

[125] *HTC Vive*. https://www.vive.com/us/. Last accessed: 2017-12-14.

[126] S.S. Hacisalihzade, L.W. Stark, and J.S. Allen. "Visual perception and sequences of eye movement fixations: A stochastic modeling approach." In: *IEEE Trans. on systems, man, and cybernetics* 22.3 (1992), pp. 474–81.

[127] H. Hadizadeh and I.V. Bajic. "Saliency-Aware Video Compression." In: *IEEE Trans. on Image Proc.* 23.1 (2014), pp. 19–33.

[128] S. Hajisharif, J. Kronander, and J. Unger. "HDR reconstruction for alternating gain (ISO) sensor readout." In: *Proc. of EUROGRAPHICS (Short Papers)*. 2014.

[129] M. Hašan, M. Fuchs, W. Matusik, H.P. Pfister, and S. Rusinkiewicz. "Physical reproduction of materials with specified subsurface scattering." In: 29.4 (2010), p. 61.

[130] V. Havran, J. Filip, and K. Myszkowski. "Perceptually motivated BRDF comparison using single image." In: *Computer Graphics Forum* 35.4 (2016), pp. 1–12.

[131] K. He, J. Sun, and X. Tang. "Guided Image Filtering." In: *IEEE Trans. Pattern Anal. Mach. Intell.* 35.6 (2013), pp. 1397–1409.

[132] L. He, M.F. Cohen, and D.H. Salesin. "The virtual cinematographer: a paradigm for automatic real-time camera control and directing." In: *Proc. of the 23rd annual conference on Computer graphics and interactive techniques*. 1996, pp. 217–224.

[133] R. Heck, M.N. Wallick, and M. Gleicher. "Virtual videography." In: *TOMCCAP* 3.1 (2007).

[134] P. Hedman and J. Kopf. "Instant 3D Photography." In: *ACM Trans. Graph.* 37.4 (2018), 101:1–101:12. ISSN: 0730-0301.

[135] P. Hedman, S. Alsisan, R. Szeliski, and J. Kopf. "Casual 3D photography." In: *ACM Trans. Graph.* 36.6 (2017), 234:1–234:15.

[136] J. Heer and M. Bostock. "Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design." In: *Proc. of ACM Conference on Human Factors in Computing Systems (CHI)*. CHI '10. Atlanta, Georgia, USA, 2010, pp. 203–212.

[137] F. Heide, W. Heidrich, and G. Wetzstein. "Fast and Flexible Convolutional Sparse Coding." In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[138] F. Heide et al. "FlexISP: A Flexible Camera Image Processing Framework." In: *ACM Trans. Graph.* 33.6 (2014).

[139] F. Heide, L. Xiao, A. Kolb, Matthias B.H., and W. Heidrich. "Imaging in scattering media using correlation image sensors and sparse convolutional coding." In: *Opt. Express* 22.21 (2014), pp. 26338–26350.

[140] Arianne T. Hinds, Didier Doyen, and Pablo Carballeira. "Toward the realization of six degrees-of-freedom with compressed light fields." In: *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*. IEEE Computer Society, 2017, pp. 1171–1176.

[141] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S.K. Nayar. "Video from a Single Coded Exposure Photograph using a Learned Over-Complete Dictionary." In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2011, pp. 287–294.

[142] J. Hochberg and V. Brooks. "Film cutting and visual momentum." In: *In the mind's eye: Julian Hochberg on the perception of pictures, films, and the world* (2006), pp. 206–228.

[143] X. Hu, Y. Deng, X. Lin, J. Suo, Q. Dai, C. Barsi, and R. Raskar. "Robust and accurate transient light transport decomposition via convolutional sparse coding." In: *Opt. Lett.* 39.11 (2014), pp. 3177–3180.

[144] J. Huang, Z. Chen, D. Ceylan, and H. Jin. "6-DOF VR videos with a single 360-camera." In: *Proc. of IEEE Virtual Reality*. IEEE Computer Society, 2017, pp. 37–44.

[145] X. Huang, C. Shen, X. Boix, and Q. Zhao. "Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks." In: 2015, pp. 262–270.

[146] R.S. Hunter and R.W. Harold. *The Measurement of Appearance (2nd Edition)*. Wiley, 1987.

[147] ITU. *ITU-R.REC.BT.500-11. Methodology for the subjective assessment of the quality for television pictures*. Tech. rep. 2002.

[148] ITU. *ITU-R.REC.P.910. Subjective audivisual quality assessment methods for multimedia applications*. Tech. rep. 2008.

[149] Intel and HypeVR Technology. *HypeVR*. `https://hypevr.com/`. Last accessed: 2018-09-14. 2018.

[150] L. Itty, C. Koch, and E. Niebur. "A model of saliency-based visual attention for rapid scene analysis." In: *IEEE Trans. PAMI* 20.11 (1998), pp. 1254–1259.

[151] E. Jain, Y. Sheikh, A. Shamir, and J. Hodgins. "Gaze-driven Video Re-editing." In: *ACM Trans. Graph.* (2014).

[152] A. Jarabo, H. Wu, J. Dorsey, H. Rushmeier, and D. Gutierrez. "Effects of Approximate Filtering on the Appearance of Bidirectional Texture Functions." In: *IEEE Trans. on Visualization and Computer Graphics* 20.6 (2014).

[153] D.S. Jeon, I. Choi, and M.H. Kim. "Multisampling Compressive Video Spectroscopy." In: *Computer Graphics Forum* 35.2 (2016).

[154] Y. Jia and M. Han. "Category-independent object-level saliency detection." In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 1761–68.

[155] M. Jiang, S. Huang, J. Duan, and Q. Zhao. "Salicon: Saliency in context." In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1072–1080.

[156] M. Jiang, X. Boix, G. Roig, J. Xu, L. Van Gool, and Q. Zhao. "Learning to predict sequences of human visual fixations." In: *IEEE Trans. on neural networks and learning systems* 27.6 (2016), pp. 1241–1252.

[157] K. Jo. "Image processing apparatus, image processing method, and program. US Patent 2014/0321766A1." Patent US 2014/0321766A1 (US). 2014.

[158] T. Judd, K. Ehinger, F. Durand, and A. Torralba. "Learning to Predict Where Humans Look." In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2009.

[159] N.K. Kalantari, T. Wang, and R. Ramamoorthi. "Learning-based view synthesis for light field cameras." In: *ACM Trans. Graph.* 35.6 (2016), 193:1–193:10.

[160] N.K. Kalantari, E. Shechtman, C. Barnes, S. Darabi, D.B. Goldman, and P. Sen. "Patch-based High Dynamic Range Video." In: *ACM Trans. Graph.* 32.6 (2013).

[161] K. Kavukcuoglu, P. Sermanet, Y. Boureau, K. Gregor, M. Mathieu, and Y.L. Cun. "Learning Convolutional Feature Hierarchies for Visual Recognition." In: *Advances in Neural Information Processing Systems 23*. Ed. by J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta. Curran Associates, Inc., 2010, pp. 1090–1098.

[162] B.W. Keelan. "ISO 20462: A psychophysical image quality measurement standard." In: *Proc. of the SPIE*. Vol. 5294. 2003, pp. 181–189.

[163] P. Kellnhofer, P. Didyk, S. Wang, P. Sitthi-Amorn, W. Freeman, F. Durand, and W. Matusik. “3DTV at Home: Eulerian-lagrangian Stereo-to-multiview Conversion.” In: *ACM Trans. Graph.* 36.4 (2017), 146:1–146:13.

[164] W.B. Kerr and F. Pellacini. “Toward Evaluating Material Design Interface Paradigms for Novice Users.” In: *ACM Trans. Graph.* 29.4 (2010), 35:1–35:10.

[165] W. Kienzle, F.A. Wichmann, M.O. Franz, and B. Schölkopf. “A nonparametric approach to bottom-up visual saliency.” In: *Proc. of Conference and Workshop on Neural Information Processing Systems (NIPS)*. 2006, pp. 689–96.

[166] H.K. Kim, J. Park, Y. Choi, and M. Choe. “Virtual reality sickness questionnaire (VRSQ): Motion sickness measurement index in a virtual reality environment.” In: *Applied Ergonomics* 69 (2018), pp. 66–73.

[167] C. Koch and S. Ullman. “Shifts in selective visual attention: towards the underlying neural circuitry.” In: *Matters of Intelligence*. Springer, 1987, pp. 115–41.

[168] K. Koehler, F. Guo, S. Zhang, and M.P. Eckstein. “What do saliency models predict?” In: *Journal of vision* 14.3 (2014), pp. 14–14.

[169] C. Kolb, D. Mitchell, and P. Hanrahan. “A realistic camera model for computer graphics.” In: *Proc. of ACM SIGGRAPH*. 1995.

[170] T. Kollenberg, A. Neumann, D. Schneider, T. Tews, T. Hermann, H. Ritter, A. Dierker, and H. Koesling. “Visual search in the (un) real world: how head-mounted displays affect eye movements, head movements and target detection.” In: *Proc. of Symposium on Eye-Tracking Research & Applications (ETRA)*. ACM. 2010, pp. 121–124.

[171] R. Koller, L. Schmid, N. Matsuda, T. Niederberger, L. Spinoulas, O. Cossairt, G. Schuster, and A.K. Katsaggelos. “High spatio-temporal resolution video with compressed sensing.” In: *Opt. Express* 23.12 (2015), pp. 15992–16007.

[172] B. Kong and C.C. Fowlkes. *Fast Convolutional Sparse Coding (FCSC)*. Tech. rep. UC Irvine, 2014.

[173] B. Koniaris, M. Kosek, D. Sinclair, and K. Mitchell. “Real-time Rendering with Compressed Animated Light Fields.” In: *Proc. of the 43rd Graphics Interface Conference*. Canadian Human-Computer Communications Society / ACM, 2017, pp. 33–40.

[174] J. Kopf, M.F. Cohen, D. Lischinski, and M. Uyttendaele. “Joint bilateral upsampling.” In: *ACM Trans. Graph.* 26.3 (2007), p. 96.

[175] Y. Koyama, D. Sakamoto, and T. Igarashi. “Crowd-powered Parameter Analysis for Visual Design Exploration.” In: *Proc. of ACM Symposium on User Interface Software and Technology (UIST)*. 2014, pp. 65–74.

[176] J. Kronander, S. Gustavson, G. Bonnet, and J. Unger. “Unified HDR reconstruction from raw CFA data.” In: *Proc. of IEEE International Conference on Computational Photography (ICCP)*. 2013, pp. 1–9.

[177] C.A. Kurby and J.M. Zacks. “Segmentation in the perception and memory of events.” In: *Trends in cognitive sciences* 12.2 (2008), pp. 72–79.

[178] Y. Lan, Y. Dong, F. Pellacini, and X. Tong. “Bi-scale appearance fabrication.” In: *ACM Trans. Graph.* 32.4 (2013), p. 145.

[179] V.P. Laurutis and D.A. Robinson. “The vestibulo-ocular reflex during human saccadic eye movements.” In: *The Journal of Physiology* 373.1 (1986), pp. 209–33.

[180] J. Lawrence, A. Ben-Artzi, C. DeCoro, W. Matusik, H. Pfister, R. Ramamoorthi, and S. Rusinkiewicz. “Inverse Shade Trees for Non-parametric Material Representation and Editing.” In: *ACM Trans. Graph.* 25.3 (2006), pp. 735–745.

[181] O. Le Meur and T. Baccino. “Methods for comparing scanpaths and saliency maps: strengths and weaknesses.” In: *Behavior research methods* 45.1 (2013), pp. 251–266.

[182] O. Le Meur, T. Baccino, and A. Roumy. “Prediction of the Inter-Observer Visual Congruency (IOVC) and application to image ranking.” In: *Proc. of ACM Multimedia*. ACM. 2011, pp. 373–382.

[183] O. Le Meur and Z. Liu. “Saccadic model of eye movements for free-viewing condition.” In: *Vision Res.* 116 (2015), pp. 152–64.

[184] H. Lee, A. Battle, R. Raina, and A.Y. Ng. “Efficient Sparse Coding Algorithms.” In: *Advances in Neural Information Processing Systems 19*. 2007, pp. 801–808.

[185] J. Lee, Y. Matsushita, B. Shi, I.S. Kweon, and K. Ikeuchi. “Radiometric Calibration by Rank Minimization.” In: *IEEE Trans. PAMI* 35.1 (2013), pp. 144–156.

[186] J.H. Lee, I. Choi, and M.H. Kim. “Laplacian Patch-Based Image Synthesis.” In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2016, pp. 2727–2735.

[187] G. Leifman, D. Rudoy, T. Swedish, E. Bayro-Corrochano, and R. Raskar. “Learning Gaze Transitions From Depth to Improve Video Saliency Estimation.” In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2017.

[188] A. Levin, D. Lischinski, and Y. Weiss. “Colorization using optimization.” In: *ACM Trans. Graph.* 23.3 (2004), pp. 689–694.

[189] A. Levin, D. Lischinski, and Y. Weiss. “A Closed-Form Solution to Natural Image Matting.” In: *IEEE Trans. Pattern Anal. Mach. Intell.* 30.2 (2008), pp. 228–242.

[190] A. Levin, R. Fergus, F. Durand, and W.T. Freeman. “Image and Depth from a Conventional Camera with a Coded Aperture.” In: *ACM Trans. Graph.* 26 (2007).

[191] G. Li and Y. Yu. “Visual saliency based on multiscale deep features.” In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 5455–63.

[192] X. Lin, Y. Liu, J. Wu, and Q. Dai. “Spatial-spectral encoded compressive hyperspectral imaging.” In: *ACM Trans. Graph.* 33 (2014), pp. 1–11.

[193] C. Lipski, C. Linz, K. Berger, A. Sellent, and M. Magnor. “Virtual Video Camera: Image-Based Viewpoint Navigation Through Space and Time.” In: *Computer Graphics Forum* 29.8 (2010), pp. 2555–2568.

[194] C. Liu. “Exploring new representations and applications for motion analysis.” PhD thesis. Massachusetts Institute of Technology USA, 2009.

[195] D. Liu, J. Gu, Y. Hitomi, M. Gupta, T. Mitsunaga, and S.K. Nayar. “Efficient Space-Time Sampling with Pixel-wise Coded Exposure for High Speed Imaging.” In: *IEEE Trans. PAMI* 36 (2013), pp. 248–260.

[196] R. Liu, J. Cao, Z. Lin, and S. Shan. “Adaptive partial differential equation learning for visual saliency detection.” In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 3866–73.

[197] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.Y. Shum. “Learning to detect a salient object.” In: *IEEE Trans. PAMI* 33.2 (2011), pp. 353–367.

[198] Z. Lu and K. Grauman. “Story-Driven Summarization for Egocentric Video.” In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 2714–2721.

[199] B. Luo, F. Xu, C. Richardt, and J.H. Yong. “Parallax360: Stereoscopic 360° Scene Representation for Head-Motion Parallax.” In: *IEEE Trans. Vis. Comput. Graph.* 24.4 (2018), pp. 1545–1553.

[200] J. Magliano and J.M. Zacks. “The Impact of Continuity Editing in Narrative Film on Event Segmentation.” In: *Cognitive Science* 35.8 (2011), pp. 1489–1517.

[201] D. Mahajan, F. Huang, W. Matusik, R. Ramamoorthi, and P. Belhumeur. "Moving Gradients: A Path-based Method for Plausible Image Interpolation." In: *ACM Trans. Graph.* 28.3 (2009), 42:1–42:11.

[202] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. "Online Dictionary Learning for Sparse Coding." In: *Proc. of International Conference on Machine Learning (ICML)*. Montreal, Quebec, Canada, 2009, pp. 689–696.

[203] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. "Online Learning for Matrix Factorization and Sparse Coding." In: *Journal of Machine Learning Research* 11 (2010), pp. 16–60.

[204] S. Malpica, M. Barrio, D. Gutierrez, A. Serrano, and B. Masia. "Improved Intuitive Appearance Editing based on Soft PCA." In: *Proc. of CEIG*. 2017.

[205] S. Malpica, A. Serrano, M. Allue, M.G. Bedia, and B. Masia. "Crossmodal Perception in Virtual Reality." In: *Multimedia Tools and Applications* (2019). To appear.

[206] A. Manakov, J.F. Restrepo, O. Klehm, R. Hegedüs, E. Eisemann, H.P. Seidel, and I. Ihrke. "A Reconfigurable Camera Add-on for High Dynamic Range, Multi-Spectral, Polarization, and Light-Field Imaging." In: *ACM Trans. Graph.* 32.4 (2013), 47:1–47:14.

[207] S. Mangiat and J.D. Gibson. "Spatially adaptive filtering for registration artifact removal in HDR video." In: *Proc. of IEEE International Conference on Image Processing (ICIP)*. 2011, pp. 1317–1320.

[208] S. Mann and R.W. Picard. "On Being undigital With Digital Cameras: Extending Dynamic Range By Combining Differently Exposed Pictures." In: *Proc. IS & T Annual Meeting*. 1995, pp. 442–448.

[209] R. Mantiuk, S. Daly, and L. Kerofsky. "Display Adaptive Tone Mapping." In: *ACM Trans. Graph.* 27.3 (2008), 68:1–68:10.

[210] R. Mantiuk, G. Krawczyk, K. Myszkowski, and H.P. Seidel. "Perception-motivated High Dynamic Range Video Encoding." In: *ACM Trans. Graph.* 23.3 (2004), pp. 733–741.

[211] R.K. Mantiuk, A. Tomaszewska, and R. Mantiuk. "Comparison of four subjective methods for image quality assessment." In: *Computer Graphics Forum* 31.8 (2012), pp. 2478–2491.

[212] J. Marco, A. Serrano, A. Jarabo, B. Masia, and D. Gutierrez. "Intuitive Editing of Visual Appearance from Real-World Datasets." In: *Proc. of Material Appearance Modeling Workshop*. 2017.

[213] G. Marmitt and A.T. Duchowski. "Modeling visual attention in VR: Measuring the accuracy of predicted scanpaths." PhD thesis. Clemson University, 2002.

[214] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar. "Compressive Light Field Photography using Overcomplete Dictionaries and Optimized Projections." In: *ACM Trans. Graph.* 32 (2013), pp. 1–11.

[215] B. Masia, A. Serrano, and D. Gutierrez. "Dynamic range expansion based on image statistics." In: *Multimedia Tools and Applications* (2015), pp. 1–18.

[216] B. Masia, A. Corrales, L. Presa, and D. Gutierrez. "Coded apertures for defocus blurring." In: *Proc. of Ibero-American Symposium in Computer Graphics*. 2011.

[217] B. Masia, L. Presa, A. Corrales, and D. Gutierrez. "Perceptually optimized coded apertures for defocus deblurring." In: 31.6 (2012), pp. 1867–1879.

[218] B. Masia, G. Wetzstein, P. Didyk, and D. Gutierrez. "A Survey on Computational Displays: Pushing the Boundaries of Optics, Computation, and Perception." In: *Computers & Graphics* 37.8 (2013), pp. 1012 –1038.

[219] W. Matusik. "A Data-Driven Reflectance Model." PhD thesis. MIT, 2003.

[220] W. Matusik, H.P. Pfister, M. Brand, and L. McMillan. "A Data-Driven Reflectance Model." In: *ACM Trans. Graph.* 22.3 (2003), pp. 759–769.

[221] W. Matusik, B. Ajdin, J. Gu, J. Lawrence, H.P.A. Lensch, F. Pellacini, and S. Rusinkiewicz. "Printing Spatially-varying Reflectance." In: *ACM Trans. Graph.* 28.5 (2009), 128:1–128:9.

[222] M.D. McCool, J. Ang, and A. Ahmad. "Homomorphic Factorization of BRDFs for High-performance Rendering." In: *Proc. of SIGGRAPH*. 2001, pp. 171–178.

[223] L. McMillan and G. Bishop. "Plenoptic Modeling: An Image-based Rendering System." In: *Proc. of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '95. 1995, pp. 39–46.

[224] A. McNamara, K. Mania, and D. Gutierrez. "Perception in graphics, visualization, virtual environments and animation." In: *ACM SIGGRAPH Asia Courses*. Hong Kong, China, 2011.

[225] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. "Equation of state calculations by fast computing machines." In: *The Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092.

[226] E. Miandji, J. Kronander, and J. Unger. "Compressive Image Reconstruction in Reduced Union of Subspaces." In: *Proc. of EUROGRAPHICS*. 2015.

[227] R. Nakashima, Y. Fang, Y. Hatori, et al. "Saliency-based gaze prediction based on head direction." In: *Vision Res.* 117 (2015), pp. 59–66.

[228] G. Nam, J.H. Lee, H. Wu, D. Gutierrez, and M.H. Kim. "Simultaneous Acquisition of Microscale Reflectance and Normals." In: *ACM Trans. Graph.* 35.6 (2016).

[229] S.K. Nayar and T. Mitsunaga. "High dynamic range imaging: spatially varying pixel exposures." In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. 2000, 472–479 vol.1.

[230] S.K. Nayar and S.G. Narasimhan. "Assorted Pixels: Multi-Sampled Imaging With Structural Models." In: *Proc. of IEEE European Conference on Computer Vision (ECCV)*. Vol. IV. 2002, pp. 636–652.

[231] R. Ng. "Fourier slice photography." In: 24.3 (2005), pp. 735–744.

[232] A. Ngan, F. Durand, and W. Matusik. "Experimental Analysis of BRDF Models." In: *Proc. of EUROGRAPHICS Symposium on Rendering*. 2005, pp. 117–126.

[233] A. Ngan, F. Durand, and W. Matusik. "Image-driven Navigation of Analytical BRDF Models." In: *Proc. of EUROGRAPHICS Symposium on Rendering*. 2006, pp. 399–407.

[234] C. Nguyen, S. DiVerdi, A. Hertzmann, and F. Liu. "Vremiere: In-headset Virtual Reality Video Editing." In: *Proc. of SIGCHI*. 2017, pp. 1–11.

[235] C.H. Nguyen, M. Kyung, J. Lee, and S. Nam. "A PCA Decomposition for Real-time BRDF Editing and Relighting with Global Illumination." In: *Proc. of EUROGRAPHICS Symposium on Rendering*. 2010, pp. 1469–1478.

[236] J.B. Nielsen, H.W. Jensen, and R. Ramamoorthi. "On Optimal, Minimal BRDF Sampling for Reflectance Acquisition." In: *ACM Trans. Graph.* 34.6 (2015), 186:1–186:11.

[237] *Nikon KeyMission*. https://www.nikonusa.com/en/nikon-products/action-cameras/index.page. Last accessed: 2017-12-14.

[238] A. Nuthmann and J.M. Henderson. "Object-based attentional selection in scene viewing." In: *Journal of Vision* 10.8 (2010), p. 20.

[239] B. O'Steen. *The Invisible Cut*. Michael Wiese Productions, 2009.

[240] *Oculus Rift*. https://www.oculus.com/rift/. Last accessed: 2017-12-14.

[241] Oculus. *Oculus Introduction to Best Practices*. https://developer.oculus.com/design/latest/concepts/book-bp/. Last accessed: 2018-09-14. 2017.

[242] R.S. Overbeck, D. Erickson, D. Evangelakos, M. Pharr, and P. Debevec. "A System for Acquiring, Compressing, and Rendering Panoramic Light Field Stills for Virtual Reality." In: *ACM Trans. Graph.* 37 (2018).

[243] N. Padmanaban, R. Konrad, T. Stramer, E.A. Cooper, and G. Wetzstein. "Optimizing virtual reality for all users through gaze-contingent and adaptive focus displays." In: *PNAS* 14 (9 2017), pp. 2183–88.

[244] J. Pan, E. Sayrol, X. Giro-i Nieto, K. McGuinness, and N.E. O'Connor. "Shallow and Deep Convolutional Networks for Saliency Prediction." In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[245] D. Parikh and K. Grauman. "Relative attributes." In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2011, pp. 503–510.

[246] N. Parikh and S. Boyd. "Proximal Algorithms." In: *Found. Trends Optim.* 1.3 (2014), pp. 127–239.

[247] J. Park and I. W. Sandberg. "Universal Approximation Using Radial-basis-function Networks." In: *Neural Comput.* 3.2 (1991), pp. 246–257.

[248] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition." In: *Proc. of Conference on Signals, Systems and Computers*. Vol. 1. 1993, pp. 40–44.

[249] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn. "Towards Foveated Rendering for Gaze-tracked Virtual Reality." In: *ACM Trans. Graph.* 35.6 (2016), 179:1–179:12.

[250] K. Pearson. "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling." In: *Philosophical Magazine* 50.5 (1900), pp. 157–175.

[251] P. Peers, D.K. Mahajan, B. Lamond, A. Ghosh, W. Matusik, R. Ramamoorthi, and P. Debevec. "Compressive Light Transport Sensing." In: *ACM Trans. Graph.* 28.1 (2009), 3:1–3:18.

[252] F. Pellacini, J.A. Ferwerda, and D.P. Greenberg. "Toward a psychophysically-based light reflection model for image synthesis." In: *Proc. of the 27th annual conference on Computer graphics and interactive techniques*. 2000, pp. 55–64.

[253] E. Penner and L. Zhang. "Soft 3D reconstruction for view synthesis." In: *ACM Trans. Graph.* 36.6 (2017), 235:1–235:11.

[254] T. Pereira and S. Rusinkiewicz. "Gamut Mapping Spatially Varying Reflectance with an Improved BRDF Similarity Metric." In: *Computer Graphics Forum* 31.4 (2012).

[255] P. Perona and J. Malik. "Scale-Space and Edge Detection Using Anisotropic Diffusion." In: *IEEE Trans. Pattern Anal. Mach. Intell.* 12.7 (1990), pp. 629–639.

[256] J. Philip and G. Drettakis. "Plane-based Multi-view Inpainting for Image-based Rendering in Large Scenes." In: *Proc. of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. I3D '18. Montreal, Quebec, Canada, 2018, 6:1–6:11.

[257] T. Portz, L. Zhang, and H. Jiang. "Random coded sampling for high-speed HDR video." In: *Proc. of IEEE International Conference on Computational Photography (ICCP)*. 2013, pp. 1–8.

[258] G. Ramanarayanan, J.A. Ferwerda, B. Walter, and K. Bala. "Visual Equivalence: Towards a New Standard for Image Fidelity." In: *ACM Trans. Graph.* 26.3 (2007).

[259] A. Ranjan, J.P. Birnholtz, and R. Balakrishnan. "Improving meeting capture by applying television production principles with audio and motion detection." In: *Proc. ACM SIGCHI*. 2008, pp. 227–236.

[260] R. Raskar. "Computational photography: Epsilon to coded photography." In: *Emerging Trends in Visual Computing* (2009), pp. 238–253.

[261] R. Raskar, A. Agrawal, and J. Tumblin. "Coded exposure photography: Motion deblurring using fluttered shutter." In: *ACM Trans. Graph.* 25 (2006), pp. 795–804.

[262] S.W. Raudensbush and A.S. Bryk. *Hierarchical Linear Models*. Sage Publications, 2002.

[263] A. Rav-Acha, Y. Pritch, and S. Peleg. "Making a Long Video Short: Dynamic Video Synopsis." In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. 2006, pp. 435–41.

[264] S. Ravishankar and Y. Bresler. "MR image reconstruction from highly undersampled k-space data by dictionaty learning." In: *IEEE Trans. on Medical Imaging* 30 (2011), pp. 1028–1041.

[265] E. Reinhard, G. Ward, S. Pattanaik, P. Debevec, W. Heidrich, and K. Myszkowski. *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting. 2nd Edition.* Morgan Kaufmann, 2010.

[266] J.R. Reynolds, J.M. Zacks, and T.S. Braver. "A Computational Model of Event Segmentation From Perceptual Prediction." In: *Cognitive Science* 31.4 (2007), pp. 613–643.

[267] C. Richardt, Y. Pritch, H. Zimmer, and A. Sorkine-Hornung. "Megastereo: Constructing High-Resolution Stereo Panoramas." In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 1256–1263.

[268] N. Riche, M. Duvinage, M. Mancas, B. Gosselin, and T. Dutoit. "Saliency and human fixations: State-of-the-art and study of comparison metrics." In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2013, pp. 1153–1160.

[269] T. Rongsirigul, Y. Nakashima, T. Sato, and N. Yokoya. "Novel view synthesis with light-weight view-dependent texture mapping for a stereoscopic HMD." In: *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*. IEEE Computer Society, 2017, pp. 703–708.

[270] M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir. "A Comparative Study of Image Retargeting." In: *ACM Trans. Graph.* 29.6 (2010), 160:1–160:10.

[271] K. Ruhland, C.E. Peters, S. Andrist, et al. "A Review of Eye Gaze in Virtual Agents, Social Robotics and HCI: Behaviour Generation, User Interaction and Perception." In: 34.6 (2015), pp. 299–326.

[272] D.D. Salvucci and J.H. Goldberg. "Identifying fixations and saccades in eye-tracking protocols." In: *Proc. of ACM Symposium on Eye-Tracking Research & Applications (ETRA)*. 2000, pp. 71–8.

[273] *Samsung Gear 360*. http://www.samsung.com/global/galaxy/gear-360/. Last accessed: 2017-12-14.

[274] A. Santella, M. Agrawala, D. DeCarlo, D. Salesin, and M. Cohen. "Gaze-based Interaction for Semi-automatic Photo Cropping." In: *Proc. of ACM SIGCHI*. 2006, pp. 771–80.

[275] E. Sayyad, P. Sen, and T. Höllerer. "PanoTrace: interactive 3D modeling of surround-view panoramic images in virtual reality." In: *Proc. of ACM Symposium on Virtual Reality Software and Technology (VRST)*. ACM, 2017, 32:1–32:10.

[276] M. Schöberl, A. Belz, A. Nowak, J. Seiler, A. Kaup, and S. Foessel. "Building a high dynamic range video sensor with spatially nonregular optical filtering." In: *Proc. of SPIE*. Vol. 8499. 2012, pp. 84990C–84990C–11.

[277] M. Schöberl, J. Keinert, M. Ziegler, J. Seiler, M. Niehaus, G. Schuller, A. Kaup, and S. Foessel. "Evaluation of a high dynamic range video camera with non-regular sensor." In: *Proc. of SPIE*. Vol. 8660. 2013, pp. 86600M–86600M–12.

[278] D.C. Schedl, C. Birklbauer, and O. Bimber. "Coded Exposure HDR Light-Field Video Recording." In: *Computer Graphics Forum* 33.2 (2014), pp. 33–42.

[279] C. Schroers, J.C. Bazin, and A. Sorkine-Hornung. "An Omnistereoscopic Video Pipeline for Capture and Display of Real-World VR." In: *ACM Trans. Graph.* 37.3 (2018), 37:1–37:13. ISSN: 0730-0301.

[280] G. Schwartz and K. Nishino. "Visual Material Traits: Recognizing Per-Pixel Material Context." In: *ICCV Workshop on Color and Photometry on Computer Vision*. 2013, pp. 883–890.

[281] S. M. Seitz and C. R. Dyer. "View Morphing." In: *Proc. of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '96. 1996, pp. 21–30.

[282] P. Sen and S. Darabi. "A Novel Framework for Imaging Using Compressed Sensing." In: *Proc. of IEEE International Conference on Image Processing (ICIP)*. 2009, pp. 2109–2112.

[283] P. Sen and S. Darabi. "Compressive Dual Photography." In: *Computer Graphics Forum* 28.2 (2009).

[284] P. Sen and S. Darabi. "Compressive Rendering: A Rendering Application of Compressed Sensing." In: *IEEE Trans. on Visualization and Computer Graphics* 17.4 (2011), pp. 487–499.

[285] P. Sen, B. Chen, G. Garg, S.R. Marschner, M. Horowitz, M. Levoy, and H. Lensch. "Dual Photography." In: *ACM Trans. Graph.* 24.3 (2005), pp. 745–755.

[286] P. Sen, N.K. Kalantari, M. Yaesoubi, S. Darabi, D.B. Goldman, and E. Shechtman. "Robust Patch-based Hdr Reconstruction of Dynamic Scenes." In: *ACM Trans. Graph.* 31.6 (2012), 203:1–203:11.

[287] A. Serrano, D. Gutierrez, and B. Masia. "An In-Depth Analysis of Compressive Sensing for High Speed Video Acquisition." In: *Proc. of IEEE International Conference on Computational Photography (ICCP), posters*. 2015.

[288] A. Serrano, D. Gutierrez, and B. Masia. "Compressive High-Speed Video Acquisition." In: *Proc. of CEIG*. 2015.

[289] A. Serrano, D. Gutierrez, K. Myszkowski, H.P. Seidel, and B. Masia. "An intuitive control space for material appearance." In: *ACM Trans. graph.* 35.6 (2016), 186:1–186:12.

[290] A. Serrano, F. Heide, D. Gutierrez, G. Wetzstein, and B. Masia. "Convolutional Sparse Coding for High Dynamic Range Imaging." In: *Computer Graphics Forum* 35.2 (2016).

[291] A. Serrano, D. Gutierrez, K. Myszkowski, H.-P. Seidel, and B. Masia. "Intuitive Editing of Material Appearance." In: *ACM SIGGRAPH 2016, posters*. 2016, 63:1–63:2.

[292] A. Serrano, V. Sitzmann, J. Ruiz-Borau, G. Wetzstein, D. Gutierrez, and B. Masia. "Movie Editing and Cognitive Event Segmentation in Virtual Reality Video." In: *ACM Trans. Graph.* 36.4 (2017).

[293] A. Serrano, I. Kim, Z. Chen, S. DiVerdi, D. Gutierrez, A. Hertzmann, and B. Masia. "Motion parallax for 360 RGBD video." In: *IEEE Transactions on Visualization and Computer Graphics* (2019). To appear.

[294] Ana Serrano, Elena Garces, Belen Masia, and Diego Gutierrez. "Convolutional Sparse Coding for Capturing High-Speed Video Content." In: *Computer Graphics Forum* (2017).

[295] Q. Shan, B. Curless, Y. Furukawa, C. Hernández, and S.M. Seitz. "Occluding Contours for Multi-view Stereo." In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2014, pp. 4002–4009.

[296] L. Sigal, M. Mahler, S. Diaz, K. McIntosh, E. Carter, T. Richards, and J. Hodgins. "A Perceptual Control Space for Garment Simulation." In: *ACM Trans. Graph.* 34.4 (2015), 117:1–117:10.

[297] A. Silverstein and J.E. Farrell. "Efficient method for paired comparison." In: *Journal of Electronic Imaging* 10.2 (2001), pp. 394–398.

[298] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein. "Saliency in VR: How do people explore virtual environments?" In: *IEEE Trans. on Visualization and Computer Graphics* 24.4 (2018), pp. 1633–1642.

[299] T.J. Smith. "The attentional theory of cinematic continuity." In: *Projections* 6.1 (2012), pp. 1–27.

[300] T.J. Smith and J.M. Henderson. "Edit Blindness: The relationship between attention and global change blindness in dynamic scenes." In: *Journal of Eye Movement Research* 2.2 (2008).

[301] A. Srikantha and D. Sidibé. "Ghost Detection and Removal for High Dynamic Range Images: Recent Advances." In: *Image Commun.* 27.6 (2012), pp. 650–662.

[302] N. Stefanoski, O. Wang, M. Lang, P. Greisen, S. Heinzle, and A. Smolic. "Automatic View Synthesis by Image-Domain-Warping." In: *IEEE Trans. Image Processing* 22.9 (2013), pp. 3329–3341.

[303] M. Stengel and M. Magnor. "Gaze-contingent Computational Displays: Boosting perceptual fidelity." In: *IEEE Signal Proc. Mag.* 33.5 (2016), pp. 139–48.

[304] M.C. Stone, W.B. Cowan, and J.C. Beatty. "Color gamut mapping and the printing of digital color images." In: *ACM Trans. Graph.* 7.4 (1988), pp. 249–292.

[305] A. Stork. "Visual Computing Challenges of Advanced Manufacturing and Industrie 4.0 [Guest editors' introduction]." In: *IEEE computer graphics and applications* 35.2 (2015), pp. 21–25.

[306] Jaunt Studios. *The Cinematic VR Field Guide - A Guide to Best Practices for Shooting 360*. https://www.jauntvr.com/cdn/uploads/jaunt-vr-field-guide.pdf. Last accessed: 2018-09-14. 2017.

[307] Y. Su and K. Grauman. "Making 360° Video Watchable in 2D: Learning Videography for Click Free Viewing." In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[308] Y.C. Su, D. Jayaraman, and K. Grauman. "Pano2Vid: Automatic Cinematography for Watching 360° Videos." In: *Proc. of Asian Conference on Computer Vision (ACCV)*. 2016.

[309] Q. Sun, L.Y. Wei, and A. Kaufman. "Mapping Virtual and Physical Reality." In: *ACM Trans. Graph.* 35.4 (2016), 64:1–64:12.

[310] T. Sun, A. Serrano, D. Gutierrez, and B. Masia. "Attribute-preserving Gamut Mapping of Measured BRDFs." In: *ACM SIGGRAPH 2017, posters*. 2017, 69:1–69:2.

[311] T. Sun, A. Serrano, D. Gutierrez, and B. Masia. "Attribute-preserving gamut mapping of measured BRDFs." In: *Computer Graphics Forum* 36.4 (2017).

[312] X. Sun, K. Zhou, Y. Chen, S. Lin, J. Shi, and B. Guo. "Interactive Relighting with Dynamic BRDFs." In: *ACM Trans. Graph.* 26.3 (2007).

[313] A. Szlam, K. Kavukcuoglu, and Y. LeCun. "Convolutional matching pursuit and dictionary training." In: *arXiv:1010.0422* (2010).

[314] J.O. Talton, D. Gibson, L. Yang, P. Hanrahan, and V. Koltun. "Exploratory Modeling with Collaborative Design Spaces." In: *ACM Trans. Graph.* 28.5 (2009), 167:1–167:10.

[315] V. Tanriverdi and R.J.K. Jacob. "Interacting with Eye Movements in Virtual Environments." In: *Proc. of ACM SIGCHI*. 2000, pp. 265–72.

[316] Unity Technologies. *Movement in VR*. https://unity3d.com/es/learn/tutorials/topics/virtual-reality/movement-vr. Last accessed: 2018-09-14. 2017.

[317] J. Thatte and B. Girod. "Towards Perceptual Evaluation of Six Degrees of Freedom Virtual Reality Rendering from Stacked OmniStereo Representation." In: *Electronic Imaging* 2018.5 (2018), pp. 352–1–352–6. ISSN: 2470-1173.

[318] J. Thatte, J. Boin, H. Lakshman, and B. Girod. "Depth augmented stereo panorama for cinematic virtual reality with head-motion parallax." In: *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*. IEEE Computer Society, 2016, pp. 1–6.

[319] M.D. Tocci, C. Kiser, N. Tocci, and P. Sen. "A Versatile HDR Video Production System." In: *ACM Trans. Graph.* 30.4 (2011), 41:1–41:10.

[320] T. Tominaga, T. Hayashi, J. Okamoto, and A. Takahashi. "Performance comparisons of subjective quality assessment methods for mobile video." In: *Proc. of 2nd. International Workshop on Quality Multimedia Experience (QoMEX)*. 2010.

[321] A. Torralba, A. Oliva, M.S. Castelhano, and J.M. Henderson. "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search." In: *Psych. Review* 113.4 (2006), p. 766.

[322] S. Tregillus, M. Al Zayer, and E. Folmer. "Handsfree Omnidirectional VR Navigation Using Head Tilt." In: *Proc. of ACM Conference on Human Factors in Computing Systems (CHI)*. 2017, pp. 4063–4068.

[323] J.A. Tropp and A.C. Gilbert. "Signal Recovery from Random Measurements Via Orthogonal Matching Pursuit." In: *IEEE Trans. on Information Theory* 53 (2007), pp. 4655–4666.

[324] A. Truong, F. Berthouzoz, W. Li, and M. Agrawala. "QuickCut: An Interactive Tool for Editing Narrated Video." In: *Proc. of ACM User Interface Software and Technology Symposium (UIST)*. 2016, pp. 497–507.

[325] E. Upenik and T. Ebrahimi. "A Simple Method to Obtain Visual Attention Data in Head Mounted Virtual Reality." In: *Proc. of IEEE International Conference on Multimedia and Expo*. Hong Kong: IEEE, 2017.

[326] P. Vangorp, J. Laurijssen, and P. Dutré. "The Influence of Shape on the Perception of Material Reflectance." In: *ACM Trans. Graph.* 26.3 (2007).

[327] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin. "Dappled Photography: Mask Enhanced Cameras for Heterodyned Light Fields and Coded Aperture Refocusing." In: *ACM Trans. Graph.* 26.3 (2007).

[328] A. Velten, D. Wu, A. Jarabo, B. Masia, C. Barsi, C. Joshi, E. Lawson, M. Bawendi, D. Gutierrez, and R. Raskar. "Femto-photography: capturing and visualizing the propagation of light." In: *ACM Trans. Graph.* 32.4 (2013), p. 44.

[329] A. Volokitin, M. Gygli, and X. Boix. "Predicting When Saliency Maps are Accurate and Eye Fixations Consistent." In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 544–552.

[330] M. Waechter, M. Beljan, S. Fuhrmann, N. Moehrle, J. Kopf, and M. Goesele. "Virtual Rephotography: Novel View Prediction Error for 3D Reconstruction." In: *ACM Trans. Graph.* 36.4 (2017). ISSN: 0730-0301.

[331] M.B. Wakin, J.N. Laska, M.F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K.F. Kelly, and R.G. Baraniuk. "Compressive Imaging for Video Representation and Coding." In: *Proc. of the Picture Coding Symposium*. 2006.

[332] L. Wang, H. Lu, X. Ruan, and M.H. Yang. "Deep networks for saliency detection via local estimation and global search." In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3183–92.

[333] T. Wang, J. Zhu, N.K. Kalantari, A. A. Efros, and R. Ramamoorthi. "Light Field Video Capture Using a Learning-based Hybrid Imaging System." In: *ACM Trans. Graph.* 36.4 (2017), 133:1–133:13.

[334] W. Wang, C. Chen, Y. Wang, T. Jiang, F. Fang, and Y. Yao. "Simulating human saccadic scanpaths on natural images." In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011, pp. 441–8.

[335] Z. Wang, E.P. Simoncelli, and A.C. Bovik. "Multi-scale Structural Similarity for Image Quality Assessment." In: *Proc. of IEEE Conf. on Signals, Systems and Computers*. 2003, pp. 1398–1402.

[336] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. “Image Quality Assessment: From Error Visibility to Structural Similarity.” In: *IEEE Trans. on Image Processing* 13.4 (2004), pp. 600–612.

[337] H.B. Westlund and G.W. Meyer. “Applying Appearance Standards to Light Reflection Models.” In: *Proc. of SIGGRAPH*. 2001, pp. 501–510.

[338] G. Wetzstein, I. Ihrke, and W. Heidrich. “Sensor Saturation in Fourier Multiplexed Imaging.” In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010.

[339] G. Wetzstein, I. Ihrke, D. Lanman, and W. Heidrich. “Computational Plenoptic Imaging.” In: *Computer Graphics Forum* 30.8 (2011), pp. 2397–2426.

[340] G. Wetzstein, D. Lanman, D. Gutierrez, and M. Hirsch. *Computational Displays*. ACM SIGGRAPH Course Notes. 2012.

[341] T. Weyrich, P. Peers, W. Matusik, and S. Rusinkiewicz. “Fabricating microgeometry for custom surface reflectance.” In: *ACM Trans. Graph.* 28.3 (2009), 32:1–32:6.

[342] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz. “High-speed videography using a dense camera array.” In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. 2004, pp. 294–301.

[343] J. Wills, S. Agarwal, D. Kriegman, and S. Belongie. “Toward a perceptual space for gloss.” In: *ACM Trans. Graph.* 28.4 (2009), 103:1–103:15.

[344] S. Wright and J. Nocedal. “Numerical optimization.” In: *Springer Science* 35 (1999), pp. 67–68.

[345] H. Wu and M. Christie. “Stylistic patterns for generating cinematographic sequences.” In: *Proc. of Workshop on Intelligent Cinematography and Editing*. 2015.

[346] G. Yu, G. Sapiro, and S. Mallat. “Solving Inverse Problems With Piecewise Linear Estimators: From Gaussian Mixture Models to Structured Sparsity.” In: *IEEE Trans. on Image Processing* 21.5 (2012), pp. 2481–2499.

[347] M. Yu, H. Lakshman, and B. Girod. “A Framework to Evaluate Omnidirectional Video Coding Schemes.” In: *Proc. of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2015.

[348] M.E. Yumer, S. Chaudhuri, J.K. Hodgins, and L.B. Kara. “Semantic Shape Editing Using Deformation Handles.” In: *ACM Trans. Graph.* 34 (4 2015), 86:1–12.

[349] J.M. Zacks. “How we organize our experience into events.” In: *Psychological Science Agenda* 24.4 (2010).

[350] J.M. Zacks and K.M. Swallow. “Foundations of attribution: The perception of ongoing behavior.” In: vol. 1. 1976, pp. 223–247.

[351] J.M. Zacks and K.M. Swallow. “Event segmentation.” In: *Current directions in psychological science* 14.2 (2007), pp. 80–84.

[352] J.M. Zacks, N.K. Speer, K.M. Swallow, and C.J. Maley. “The brain’s cutting-room floor: Segmentation of narrative cinema.” In: *Frontiers in human neuroscience* 4 (2010), p. 168.

[353] M.D. Zeiler, G.W. Taylor, and R. Fergus. “Adaptive deconvolutional networks for mid and high level feature learning.” In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2011, pp. 2018–2025.

[354] M.D. Zeiler, D. Krishnan, G.W. Taylor, and R. Fergus. “Deconvolutional Networks.” In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010, pp. 2528–2535.

[355] E. Zell, C. Aliaga, A. Jarabo, K. Zibrek, D. Gutierrez, R. McDonnell, and M. Botsch. “To Stylize or Not to Stylize?: The Effect of Shape and Material Stylization on the Perception of Computer-generated Faces.” In: *ACM Trans. Graph.* 34.6 (2015), 184:1–184:12.

[356] Q. Zhao and C. Koch. "Learning saliency-based visual attention: A review." In: *Signal Proc.* 93.6 (2013), pp. 1401–7.

[357] R. Zhao, W. Ouyang, H. Li, and X. Wang. "Saliency detection by multi-context deep learning." In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1265–74.

[358] C. Zhou, S. Lin, and S.K. Nayar. "Coded Aperture Pairs for Depth from Defocus." In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2009, pp. 325–332.

[359] C. Zhou and S. Nayar. "What are good apertures for defocus deblurring?" In: *Proc. of IEEE International Conference on Computational Photography (ICCP)*. 2009, pp. 1–8.

[360] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely. "Stereo Magnification: Learning View Synthesis Using Multiplane Images." In: *ACM Trans. Graph.* 37.4 (2018), 65:1–65:12. ISSN: 0730-0301.

[361] T. Zickler, R. Ramamoorthi, S. Enrique, and P.N. Belhumeur. "Reflectance sharing: Predicting appearance from a sparse set of images of a known shape." In: *IEEE Trans. PAMI* 28.8 (2006), pp. 1287–1302.

[362] H. Zimmer, A. Bruhn, and J. Weickert. "Freehand HDR Imaging of Moving Scenes with Simultaneous Resolution Enhancement." In: *Computer Graphics Forum* 30.2 (2011), pp. 405–414.

[363] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S.A.J. Winder, and R. Szeliski. "High-quality video view interpolation using a layered representation." In: *ACM Trans. Graph.* 23.3 (2004), pp. 600–608.

[364] C.J. Zubiaga, A. Muñoz, L. Belcour, C. Bosch, and P. Barla. "MatCap Decomposition for Dynamic Appearance Manipulation." In: *Proc. of EUROGRAPHICS Symposium on Rendering (Experimental Ideas & Implementations)*. 2015.