Rendering of HDR 3D point clouds

Ific Goudé

UNIVERSITÉ DE RENNES 1

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES 1

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par

Ific GOUDÉ

Rendering of High Dynamic Range 3D point clouds

Thèse présentée et soutenue à Rennes, le 27/05/2021
Unité de recherche : Univ Rennes, CNRS, IRISA

Rapporteur.se.s avant soutenance :

Raphaëlle CHAINE	Professeure, Université Claude Bernard Lyon 1
Daniel MENEVEAUX	Professeur, Université de Poitiers

Composition du Jury :

Président :	Éric MARCHAND	Professeur, Université de Rennes 1
Examinateur.rice.s :	Raphaëlle CHAINE	Professeure, Université Claude Bernard Lyon 1
	Daniel MENEVEAUX	Professeur, Université de Poitiers
	Céline LOSCOS	Professeure, Université de Reims Champagne Ardenne
	Nicolas MELLADO	Chargé de recherche, CNRS, Université Paul Sabatier
Dir. de thèse :	Rémi COZOT	Professeur, Université du Littoral Côte d'Opale
Co-dir. de thèse :	Kadi BOUATOUCH	PU Emerite, Université de Rennes 1

Acknowledgements

First, I would like to thank Rémi Cozot for having trusted me from the beginning. Thank you for the astronomy books, the oculars, for having shared with me your passion for imaging, for all of our conversations (sometimes technical, sometimes philosophical) around coffees and cigarettes. I also wanted to thank Olivier Le Meur for having accepted to be my adoptive supervisor, for your generosity, your humanity, and your pedagogy. Thank you for all of your so clear and pragmatic advice. Special thanks to Kadi Bouatouch, my last supervisor, for those so long hours you lost at my side, correcting the papers and teaching me English, Arabic, Spanish, Portuguese, etc. For all your honest and helpful advice, your kindness, and your love for people.

Thanks to all jury members for having accepted evaluating my work, for all your compliments about the manuscript and the defense, for the interesting questions and discussions following the defense. Special thanks to Éric Marchand who is gravitating around my educational and career paths for a long time. Thanks to the members of my CSI: David Vanderhaeghe, Fabric Lamarche and Éric Marchand. Many thanks to all current and formal members of the Percept team for the friendly working atmosphere, for the coffee breaks, and for aperitifs. Special dedication to Alexandre, the best office colleague. A short digression to thank Jean-Christophe and Fabrice for having welcomed me in your Java gang, I had really good times with you (even at 8 am).

I also would like to thank some people that have accompanied me during these three years. Thanks to Thomas and Jeremy, you helped me a lot before and during the thesis. I almost forgot to mention Furtys, who could kill me if I did not thank him. Thousand thanks to Vincent (my mate) and the awesome Hybrid team for having hosted me for lunch during my first year and for your kindness, especially Florian and Rebecca. Thanks to my friends Martin, Lucas, and Vincent for the "midi entre potes". Thanks to Alexis, who was away from me, but always in my heart. Thanks to all of my friends from the IUT: Adrien, Anthony, Arthur, Ewen, Hugo, Kévin, Mathilde, Morgane, and Thibault. Thanks to my old friends Emeric, Pierre and Guillaume. Thanks to Romain for having shared your "thesis pain" and for the organization of the 3B for the old b-com team. Thank you again, Vincent, for the defense's poster of exceptional quality, and thank you Alexandre for having managed the stream with talent during the defense. Thanks to all my family, especially to my uncle Rocky who taught me to ask questions and to think.

Finally, thanks to my best friend, my girlfriend, my right-hand Solenn for having supported me for 11 years (particularly these three last years, I have not always been very cool), thank you so much!

Résumé

Introduction

Ces dernières années, l'émergence des technologies de capture et d'affichage à Grande Gamme Dynamique (HDR) ont permis de grandement améliorer la qualité des images numériques. D'un autre côté, la multiplication des capteurs chez les constructeurs de smartphone semble être une solution intéressante afin d'améliorer la qualité des images capturées. Le projet ReVeRY, dans lequel est impliquée cette thèse, cherche à anticiper les besoins de demain dans le domaine de l'imagerie en proposant de capturer un nuage de points 3D à Grande Gamme Dynamique à l'aide d'une grille de caméras. Dans ce projet, des études sur la qualité perçue, la compression ou encore l'acquisition de tels nuages de points ont été menées. Les contributions de cette thèse se concentrent sur la restitution visuelle de ces nuages de points 3D HDR.

L'aspect géométrique en 3 dimensions permet une plus grande flexibilité dans l'édition et l'interaction avec le contenu tandis que le HDR offre une représentation plus fidèle de l'intensité lumineuse et des couleurs de la scène. L'objectif principal est de surmonter les difficultés à restituer un nuage de points 3D HDR sur des écrans aux caractéristiques variées, et l'enjeu est double : 1) Le rendu de contenu 3D HDR sur des écrans classiques est une tâche difficile. Les caméras HDR capturent toute la dynamique de la scène avec beaucoup de détails dans les zones très claires et très sombres, produisant des images réalistes. La recherche autour des images HDR a connu de grandes avancées ces dernière années, néanmoins très peu de travaux ont été réalisé concernant l'affichage de ce genre de contenu 3D HDR. Nos contributions dans ce domaine ont permis d'améliorer la qualité du rendu de ces contenus lorsqu'ils sont affichés sur des écrans classique ou dans des casques de Réalité Virtuelle (VR). 2) L'absence de méthode de stylisation de contenu 3D. Alors qu'il existe beaucoup de techniques pour styliser des images (filtres, effets de flou ou de vignette, etc.), la stylisation du contenu 3D a été très peu traité. Nous présentons notre méthode de transfert de couleur de nuage de points basée sur l'exemple, qui prend en compte la géométrie. Nos résultats et évaluations ont montrés une amélioration significative par rapport aux méthodes de transfert de couleur existantes.

Contexte

Les technologies à **Grande Gamme Dynamique** cherchent à reproduire ce que peut percevoir l'œil humain. Depuis les lumières très faible des étoiles dans la nuit jusqu'à la lumière éblouissante du soleil, le Système Visuel Humain (HSV) peut voir une très grande gamme dynamique. Néanmoins, les caméras et les écrans classiques, dit à "Gamme Dynamique Standard" (SDR) ne sont pas capable de capturer et d'afficher toute la grande gamme dynamique que nous pouvons percevoir. Récemment, quelques caméras et écrans HDR sont arrivés sur le marché, rendant cette technologie disponible au grand publique. De plus, certains médias, tel que Netflix ou Amazon Prime, distribuent d'ores et déjà des films HDR sur leurs plateformes, permettant aux propriétaires d'écrans HDR de profiter du contenu dans sa plus haute qualité. D'un autre côté, les propriétaires d'écrans SDR doivent eux aussi être en mesure de regarder le contenu HDR. Afin d'afficher des images HDR sur des écrans SDR, la gamme dynamique des images est alors comprimée afin de s'adapter à celle des écrans. Cette opération est assurée par un Opérateur de Mappage des Tons (TMO) qui cherche à préserver au mieux les contrastes globaux et locaux des images HDR d'origine. La recherche dans le domaine des TMOs a débuté il y a de nombreuses années et a donné lieu à différentes méthodes. Les TMOs existants sont souvent basés sur la perception de l'œil humain afin d'assurer un traitement d'image cohérent avec la façon dont l'image tone mappé sera perçue. Comme la perception diffère en fonction des conditions de visualisation, les TMOs doivent être adaptés à l'écran sur lequel le contenu est affiché. Dans cette thèse, nous nous sommes intéressés à l'affichage de contenu HDR dans des casques de VR.

Les **nuages de points** sont étudiés depuis des décennies et commencent à devenir un format couramment utilisé dans la recherche et l'industrie. L'émergence des nouveaux scanners type LiDAR ont rendus l'utilisation des nuages de points d'une absolue nécessité. Ils sont utiles pour capturer l'environnement extérieur des véhicules autonomes, scanner les sculptures afin de préserver notre héritage culturel, obtenir les modèles 3D de bâtiments pour la construction et la rénovation et bien d'autre encore. Dans l'informatique graphique, les modèles 3D sont représentés soit par des nuages de point, soit par des maillages. Comme les maillages sont composés d'ensembles de triangles connectés par leurs sommets, supprimer cette connectivité résulte en un ensemble de points dans l'espace 3D. Les nuages de points sont donc une représentation plus générale des modèles 3D. De plus, un nombre croissant de nuages de points sont générés en scannant des objets du monde réel. En outre, plusieurs travaux de recherche se sont penchés sur le problème de la fidélité

des rendus de nuages de points. Parmi ces problèmes on peut citer le bouchage de trous, la suppression du crénelage des bords ou encore le dessin des silhouettes. Dans cette thèse, nous avons adressé le problème du transfert de couleur basé sur l'exemple.

Contributions

1) L'étude de la perception de l'intensité lumineuse et des couleurs dans les casques de réalité virtuelle. Avoir une meilleure compréhension du système visuel humain et de comment il réagit aux stimuli est essentiel pour effectuer un TMO efficace. De nombreuses recherches sur le HVS ont permis de développer des modèles mathématiques qui décrivent l'apparence de la couleur d'un stimulus et comment celui-ci est perçu par l'œil humain, en fonction des conditions de visualisation telles que l'intensité lumineuse du fond ou l'éclairage de l'environnement. Ces modèles, connus sous le nom de Modèle d'Apparence des Couleurs (CAM), ont été définit grâce à des études subjectives menées sur des écrans 2D SDR. D'un autre côté, dû à la croissance des technologies de réalité virtuelle ces dernières années, la visualisation des images à 360° dans les casques de VR est devenue courante. La grande majorité des casques manufacturés sont encore dotés d'écrans SDR, les TMOs sont donc essentiels pour afficher toute la grande dynamique des images HDR. Néanmoins, aucun CAM n'existe pour satisfaire les conditions de visualisation spécifique rencontrées dans les casques de VR. Notre première contribution fut de mener des évaluations subjectives sur la perception de l'intensité lumineuse et des couleurs dans les casques de VR et de proposer un CAM adapté à la perception de l'œil humain dans ces casques. Ces études ont été publié dans un numéro spécial du journal *Transactions on Computational Science*. En plus de cette première contribution (proposer un CAM adapté aux casques de VR), effectuer le TMO d'un contenu 3D sans le considérer dans son ensemble peut conduire à des artéfacts de cohérence spatial et temporel. Afin d'aborder ce problème, nous avons développé deux TMOs qui seront détaillés ci-après.

2) TMO d'un nuage de points 3D HDR depuis un point de vue fix. Premièrement, nous avons décidé d'étudier le TMO d'un nuage de points 3D HDR visualisé depuis une position de caméra fixée. Tandis que la position de la caméra est fixe, la caméra peut tourner, faisant potentiellement face à toutes les zones de la scène dans toutes les directions. Une situation similaire survient lors de la visualisation d'une image HDR à 360° dans un casque de VR, où l'utilisateur peut regarder toutes les parties de l'image omnidirectionnelle qui l'entoure. Pour répondre aux problèmes de cohérence spatiale, nous avons développé un TMO dédié aux images HDR à 360° visualisées dans les casques de

VR. Ces travaux ont été présentés lors de la conférence *Computer Graphics International* qui a eu lieu à Calgary en 2019. La prochaine étape de nos recherches fut de supprimer la contrainte de positionnement de la caméra et de considérer un déplacement libre dans un nuage de points 3D HDR.

3) TMO d'un nuage de points 3D HDR en déplacement libre. Délaisser un point de vue fix de la caméra permet d'effectuer un rendu du nuage de points HDR depuis tous les points de vue possible. Comparé à notre précédente contribution, permettre un déplacement complètement libre dans le contenu apporte de nouvelles difficultés. Proposer un TMO d'une séquence d'images HDR tout en préservant la cohérence globale du nuage de points 3D est difficile. Le principal problème étant d'estimer efficacement la gamme dynamique globale de la scène en temps réel alors que cette gamme dynamique peut grandement évoluer au cours du temps. Nous avons alors proposé un nouveau TMO temps-réel dédié aux scènes 3D HDR interactives. Ce TMO est efficace aussi bien pour les nuages de points HDR que pour les scènes 3D plus conventionnelles composées de maillages. Ces travaux ont donné lieu à une publication dans le journal *Computer and Graphics*. Après avoir proposé une solution satisfaisante pour la visualisation des nuages de points 3D HDR sur des écrans SDR, nous nous sommes penchés sur la problématique de rendu stylisé des nuages de points. La prochaine partie détaille les challenges et les contributions que nous avons apportés dans ce domaine.

4) Transfert de couleur pour des nuages de points 3D basé sur l'exemple. L'édition de nuages de points 3D plutôt que d'images 2D offre de plus grandes possibilités et produit de meilleurs résultats. Afin de prouver l'efficacité de travailler sur les nuages de points directement, nous avons abordé la stylisation de ces nuages de points basé sur des méthodes de transfert de couleur. De la même manière que le transfert de couleur entre des images, un premier nuage de points d'entré représente le contenu, et un second nuage de points cible représente le style. Nous souhaitons transférer le style de la cible sur l'entré tout en préservant sa géométrie. Notre méthode de transfert de couleur se repose sur les distributions des couleurs et prend en compte la géométrie des nuages de points afin de produire un résultat cohérent. Plus précisément, nous proposons deux méthodes qui font différentes hypothèses sur la forme des distributions des couleurs des nuages de points. Ensuite, le transfert de couleur est appliqué en fonction de la corrélation entre les couleurs et les normals des points. Ces travaux ont été soumis au journal *Computer Graphics Forum* et seront probablement acceptés après une révision mineure.

Table of Contents

Chapter 1

Introduction

In the imagery research field, recent years have seen the emergence of High Dynamic Range (HDR) technology. Thanks to HDR imaging, the quality of digital images has been vastly improved, offering better contrasts and much more details. However, different approaches are still investigated to improve the image quality. An interesting research direction is to use different types of sensors. For example, Apple added a LiDAR on the last iPhone to improve the image quality with depth data while other constructors multiply the number of cameras on their smartphones. The project (ReVeRY) in which this thesis is involved aims to anticipate tomorrow's need in the imagery domain by proposing the capture of a 3-dimensional HDR point cloud using a grid of cameras. Moreover, perceived quality, compression, and acquisition of such point clouds are emerging works. As detailed later, the ReVeRY project tackles all these issues, our contribution is concerned with the visual restitution of the HDR 3D point cloud.

To introduce the purpose of the thesis, we present the context and the objectives as well as particular notions like HDR and point clouds stylization. Then, we discuss the main challenges and the contribution we made before presenting the manuscript outline.

1.1 Rendering of High Dynamic Range 3D point clouds

The work of this thesis aims to improve the quality of advanced multimedia, and particularly the new 3D and HDR video format, from editing to rendering. The 3-dimensional geometric aspect of such content allows more editions and interactions while the High Dynamic Range better reproduces the light intensity and the colors of the scene. This work is supported by the ANR ReVeRY project [1] whose objective is to offer content creators a richer video format pipeline, from capturing to rendering. This project consists of three steps. First, a grid of cameras is designed to capture a scene from several viewpoints with several exposures (see Figure 1.1). Then, this multi-view and multi-exposed video stream

1. ANR project ANR-17-CE23-0020 (`https://revery.univ-reims.fr/`).

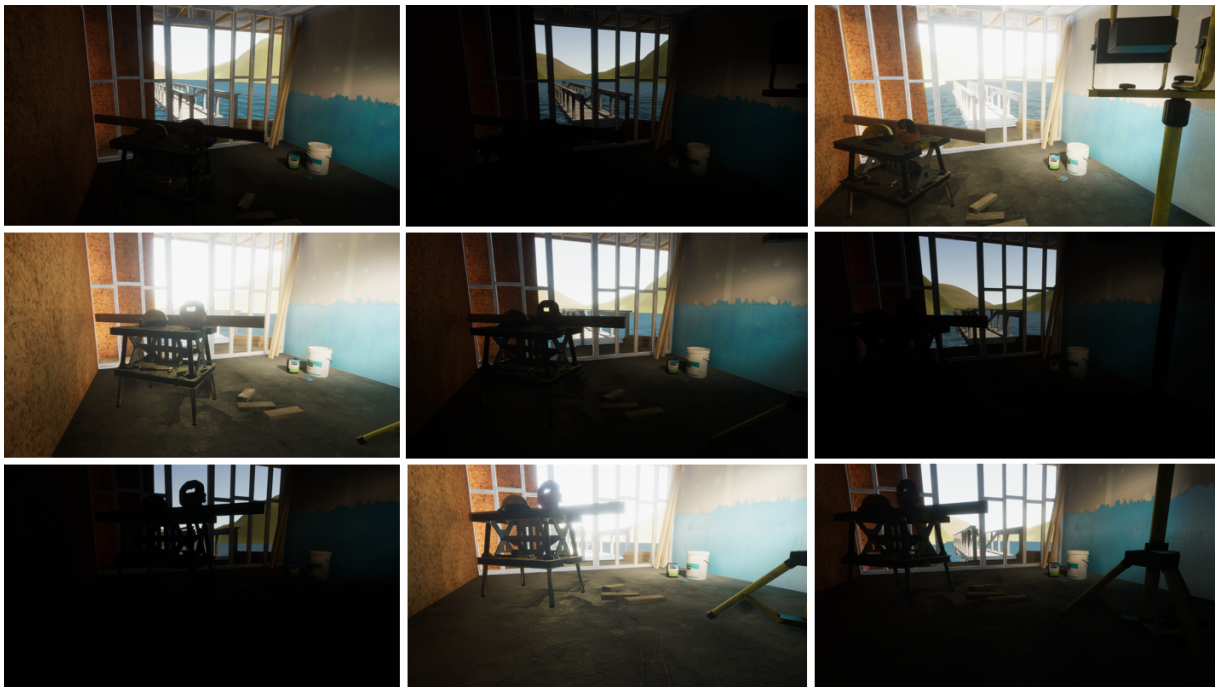


Figure 1.1: Multi-view and multi-exposed grid of cameras output.

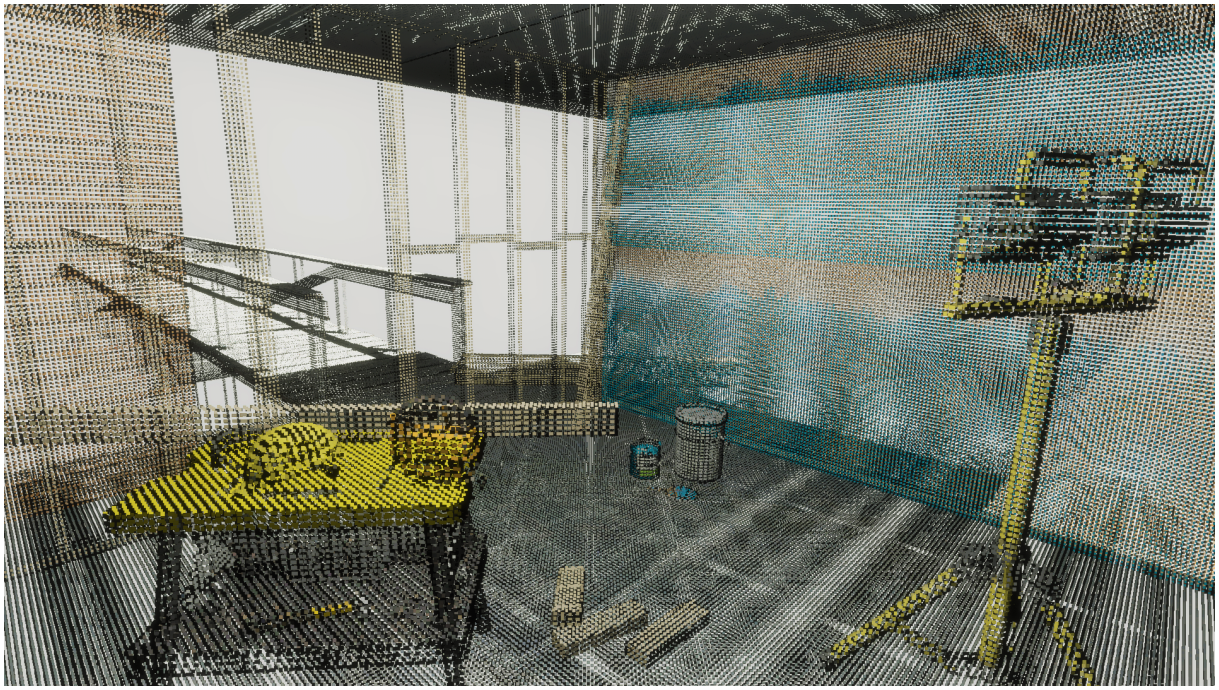


Figure 1.2: Reconstructed HDR 3D point cloud.

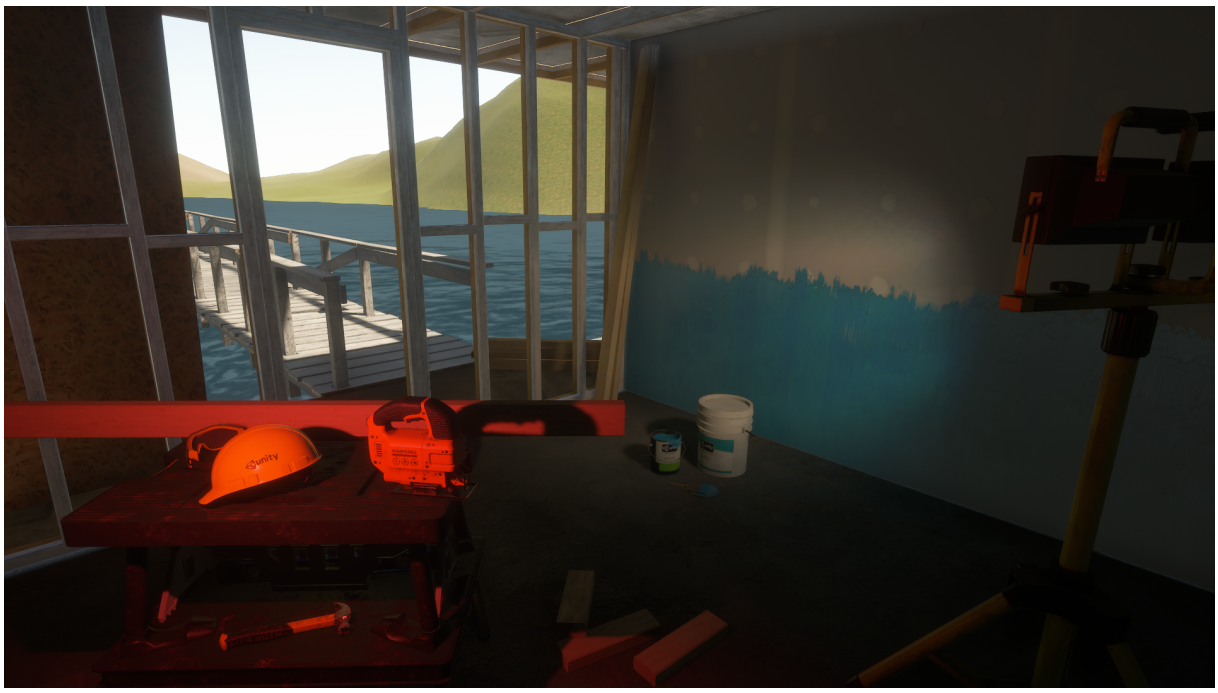


Figure 1.3: Final rendering of the HDR 3D point cloud with a viewpoint change and post-processing effects.

is converted into an Ultra High Definition HDR 3D point cloud video (see Figure 1.2). Finally, this HDR 3D point cloud video is edited and rendered on a display, which can be 2D, 3D, stereoscopic, HDR, etc. (see Figure 1.3).

This last step is the thesis' central object. The main goal is to overcome the difficulties of rendering HDR 3D point clouds on multiple displays with a huge variety of features. The rendering process is the visual restitution of data onto a screen. It may be of different nature, from displaying raw data for professional uses to lighting simulations for image synthesis and so on. In this thesis we address the rendering of HDR 3D point clouds in two ways:

1. Rendering an HDR 3D content on a classical display. While HDR cameras capture the whole dynamic of a scene with a lot of details in the brightest and darkest areas, classical displays have a Standard Dynamic Range (SDR), preventing them from displaying HDR contents without losing details. To appreciate such HDR contents through SDR displays, the process of tone mapping is used to adapt HDR images to the range of these SDR displays.

2. Producing a stylized rendering of a point cloud. While it exists a huge amount

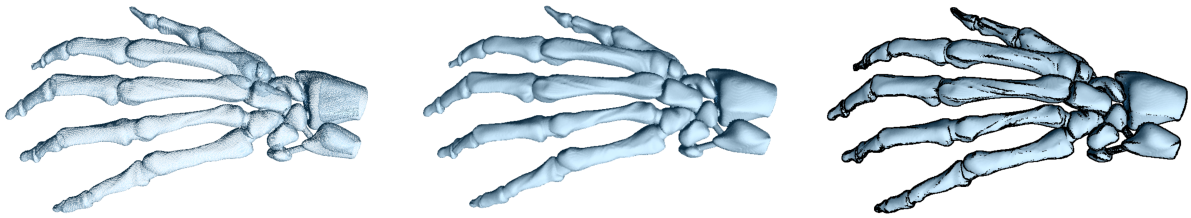
of style effects for images (filtering, blurring, vignetting, etc.), only a few works exist for applying style effects to 3D contents. We focused on example-based color transfer accounting for point clouds geometry.

High Dynamic Range technologies aim to reproduce what the human eyes perceive. From the weak light of stars in the night to the bright light of the sun, the Human Visual System (HVS) can see a very high dynamic range that befalls in the real world. However, regular SDR cameras and screens are not able to capture and display this perceptible high dynamic range, which results in a loss of details in bright and dark areas as illustrated in Figure 1.4. Recently, a few HDR cameras and HDR displays have come onto the market and became publicly available. Indeed, media streamers (like Netflix or Amazon Prime) are already distributing HDR movies on their platforms, allowing HDR screens owners to watch such content with high quality. On the other hand, SDR screens owners should be able to watch HDR content too. To render HDR images on SDR screens, the dynamic range of images is compressed to fit with that of screens. This operation is ensured by a Tone Mapping Operator (TMO) that seeks to preserve as best as possible global and local contrasts of the original HDR image (see Figure 1.5). Research in the field of TMOs started many years ago and has given rise to many different methods. The main purpose is to visualize HDR images onto an SDR screen. It amounts to a compression problem with loss, that seeks to preserve global and local contrasts as well as maximum details. Existing TMOs are often based on human eye perception to ensure the image processing to be coherent with the way the tone-mapped image will be perceived. As the perception differs depending on the visual conditions, TMOs have to be adapted to the screen the content is displayed on. For the thesis purpose, we are interested in displaying HDR content on Head-Mounted Displays (HMDs).

Figure 1.4: SDR images are very limited and are not representative of what the human eyes perceive. *Images courtesy of Mertens et al. [1].*

Figure 1.5: Tone mapping the High Dynamic Range of the image preserving global and local contrasts. *Image courtesy of Mertens et al. [1].*

Point clouds are now studied for decades and start to be a regular format used in research and industry. The emergence of new scanners, like LiDAR technologies, helps a lot in making point cloud data format necessary. It is useful for capturing the surrounding environment of autonomous vehicles, scanning sculptures to preserve the cultural heritage [2], [3], obtaining 3D models of buildings for renovations and constructions [4], [5], and so on. In computer graphics, 3D models are represented either by point clouds or by meshes. As meshes are made of a set of triangles connected through vertices, removing this connectivity results in a set of points in a 3D space. Point clouds are a more general representation of 3D models. Indeed, an increasing number of 3D point clouds are generated by scanning real-world objects. Besides, several papers have tackled problems of rendering point clouds faithfully. These problems are holes filling, edges aliasing removal, or silhouette drawing as illustrated in Figure 1.6. In this thesis, we have addressed



(a) Original point cloud. (b) Holes filling. (c) Silhouette drawing.

Figure 1.6: Improving the rendering quality of point clouds. *Images courtesy of Rosenthal and Linsen [6].*

example-based color transfer between point clouds. Our contributions to the rendering of HDR 3D point clouds are detailed in the following sections.

1.1.1 Lightness and color perception on Head Mounted Display

Understanding the Human Visual System (HVS) and how it reacts to stimuli is essential to ensure an efficient tone mapping. Many research works on the HVS lead to develop mathematical models to describe the color appearance of a stimulus and how it is perceived by the human eyes, depending on viewing conditions such as the luminance of the background and the surround. These models, known as Color Appearance Models (CAMs), have been defined thanks to psychophysical studies conducted on SDR 2D screens. Furthermore, due to the growth of Virtual Reality (VR) technologies over the last years, the visualization of 360° images on Head-Mounted Displays (HMDs) has become common. The huge majority of manufactured HMDs still having SDR screens, TMOs are then essential to display all the dynamic range of HDR images. Nevertheless, no CAM exists for satisfying the specific visualization conditions encountered on HMDs. Our first contribution, detailed in Chapter 5, is to conduct subjective evaluations on lightness and color perception on HMDs and to propose an adapted CAM that fits with the human eye perception on such HMDs. These studies have been published in a special issue of Transactions on Computational Science journal [7]. In addition to this first contribution (i.e. having a CAM adapted to HMDs), tone mapping a 3D content, without considering it in its entirety, may lead to spatial and temporal coherency artifacts. To tackle these issues, we developed two TMOs that are detailed in the next sections.

1.1.2 Tone mapping High Dynamic Range 3D point clouds from a fixed viewpoint

As a first step, we decided to investigate the tone mapping of HDR 3D point clouds visualized from a fixed camera position. Whereas the position of the camera is fixed, the camera can rotate, potentially facing all areas of the scene from every direction. A similar situation occurs when visualizing a 360° HDR image on HMD, where the user can watch any parts of the surrounding omnidirectional image. To answer spatial coherency issues, we developed a TMO dedicated to 360° HDR images visualized on HMD as described in Chapter 6. This work has been presented during the Computer Graphics International conference which was held in Calgary in 2019 [8]. The next step of our research was

to remove the constraints in camera positioning and to consider a free walk-through interactive HDR 3D point cloud.

1.1.3 Tone mapping High Dynamic Range 3D point clouds in a free walk-through

Relaxing the fixed camera position allows rendering HDR point clouds from every possible viewpoint in the scene. Compared to our contribution presented in the previous subsection, allowing completely free walk-through results in new difficulties. Tone mapping the rendered HDR image sequence while preserving a global coherency of the HDR 3D point clouds is challenging. The principal issue was to efficiently estimate the global dynamic range of the whole scene in real-time while this dynamic range can significantly change over time. We then propose a new real-time TMO dedicated to HDR rendering of interactive 3D scenes. This TMO is efficient either for HDR 3D point clouds or for more conventional 3D scenes composed of meshes as explained in Chapter 7. This work gave rise to a publication in the Computer and Graphics journal [9]. Whereas a satisfying solution for visualizing HDR 3D point clouds on SDR screens has been found, we addressed the problem of rendering a stylized point cloud. The next section details the challenges and the contributions we brought to this domain.

1.1.4 Example-based color transfer for 3D point clouds

Editing 3D point clouds instead of images offers much more possibilities and produces better results. To prove the efficiency of working on point clouds directly, we tackled the stylization of such point clouds based on color style transfer. With a similar process to image color transfer, an input point cloud represents the content, and a target point cloud represents the style. We want to transfer the color style of the target point cloud to the input point cloud by preserving its geometry. Our color transfer method relies on color distributions and takes into account the point clouds geometry to produce a coherent result. More precisely, we propose two methods that make different assumptions regarding the shape of the distributions of the point clouds colors. Then, the color transfer is performed depending on the correlations between the colors and the normals of the point clouds. This work, presented in Chapter 8, has been submitted to the Computer Graphics Forum journal.

1.2 Summary of contributions

In this thesis, our contributions in the field of HDR 3D point clouds rendering are:

1. a CAM adapted to the viewing conditions on HMD that relies on two subjective studies for lightness and color perception on HMD;
2. a TMO dedicated to the visualization of 360° HDR images on HMD which preserves the spatial coherency of the scene in addition to relying on our CAM for HMD to produce perceptually coherent tone mapped images;
3. a real-time TMO dedicated to free walk-through interactive HDR 3D scene which preserves both spatial and temporal coherency;
4. a subjective evaluation protocol to compare HDR content visualized on an HDR 2D display with its tone-mapped version visualized on an HMD;
5. two methods for transferring the colors of a point cloud to another by accounting for the point clouds geometry;
6. objective metrics to evaluate the quality of the color transfer between two point clouds.

1.3 List of publications

These contributions, previously enumerated, have led to the submission and the publication of the four articles cited below:

1. **HMD-TMO: A Tone Mapping Operator for 360° HDR Images Visualization for Head Mounted Displays**, Goudé Ific, Cozot Rémi and Banterle Francesco, *Computer Graphics International Conference*, 216-227 (2019)
2. **A Perceptually Coherent TMO for Visualization of 360° HDR Images on HMD**, Goudé Ific, Cozot Rémi and Le Meur Olivier, *Transactions on Computational Science XXXVII*, 109-128 (2020)
3. **Tone mapping High Dynamic 3D scenes with global lightness coherency**, Goudé Ific, Lacoche Jérémy and Cozot Rémi, *Computer & Graphics, Volume 91*, 243-251 (2020)
4. **Example-based stylization of 3D point clouds**, Goudé Ific, Cozot Rémi, Le Meur Olivier and Bouatouch Kadi, *Computer Graphics Forum* (minor revision)

1.4 Organization of the thesis

The manuscript is divided into four main parts as follows:

— **Part I - Background:** we start by giving the elementary notions useful for understanding this thesis. In Chapter 2 we present the basis of human eye perception as well as the color appearance models. In Chapter 3 we present the state of the art regarding HDR imaging and tone mapping. Finally, in Chapter 4, we formally introduce the point cloud data format and the way of rendering such point clouds.

— **Part II - Human eye perception on Head Mounted Display:** In Chapter 5 we start by explaining our scientific approach regarding human eye perception before detailing the subjective studies we conducted about HMD visualization. This chapter concludes with the definition of our CAM dedicated to HMD.

— **Part III - Tone Mapping High Dynamic Range 3D point cloud:** First, in Chapter 6, we describe our TMO dedicated to the visualization of 360° HDR images on HMD. Our TMO relies on the combination of two TMOs: 1) a first one applied to the current viewport in real-time that enhances the image contrast and 2) a second applied to the whole 360° image that preserves the global coherency while being perceptually coherent with the visualization on HMD. Compared to existing TMOs, our method obtained better quality scores overall a set of 90 images. Second, in Chapter 7, we present our TMO dedicated to interactive walk-through in HDR 3D scenes. This second TMO is adapted from the previous one, a combination of a viewport and a global TMO, whereas the main contribution focuses on the definition of the dynamic range of the 3D scene. We conducted a subjective study to evaluate the performance of our TMO compared to existing ones and found that our method is globally more appreciated than the others.

— **Part IV - Point cloud stylization:** Our work on rendering stylized point clouds is presented in Chapter 8. We detail our pipeline for transferring colors between point clouds considering their geometry. We proposed to adapt two color transfer methods from the image domain that rely on fitting the color distributions of the two images, assuming that the colors follow Gaussian distributions. Moreover, we proposed several objective metrics to evaluate the quality of the color transfer between point clouds and found that our methods obtain the best scores overall.

1.5 Glossary

HDR	High Dynamic Range
SDR	Standard Dynamic Range
HMD	Head-Mounted Display
TMO	Tone Mapping Operator
VR	Virtual Reality
HVS	Human Visual System
CIE	International Commission on Illumination
CAM	Color Appearance Model
PBR	Physically Based Rendering
JND	Just Noticeable Difference
FoV	Field of View
CDF	Cumulative Distribution Function
MGD	Multivariate Gaussian Distribution
MGGD	Multivariate Generalized Gaussian Distribution
CNN	Convolutional Neural Network
PCA	Principal Component Analysis
EMD	Earth Mover's Distance

Part I

Background

INTRODUCTION

This part introduces theoretical concepts regarding color perception (Chapter 2), High Dynamic Range imaging as well as tone mapping (Chapter 3) and point cloud rendering (Chapter 4). Understanding these notions is essential before delving into the main body of the thesis.

Chapter 2

Human Visual System

This chapter introduces the functioning of the HVS and the basis of color spaces. The goal is not to provide a thorough and comprehensive review of the HVS. Readers could refer to [10] to get an in-depth review about visual functioning and color appearance. Notions needed to understand the following of the thesis (such as relative perception and temporal adaptation) are presented below with their definition of some useful terms.

2.1 Physical functioning

Light is what we perceive from the world through our eyes. Waves, of photon particles emitted and reflected by objects, scatter in the world until reaching the photoreceptor cells in our eyes. These cells, responsible for our visual perception, are named rods and cones. Rods are sensitive to low luminance levels (referred to as scotopic vision) while cones are sensitive to high luminance levels (referred to as photopic vision). The vision where both rods and cones are active is referred as mesopic vision as illustrated in Figure 2.1. Moreover, cones are in charge of our perception of color. There are three types of cones sensitive to specific wavelengths of the visible color spectrum as showed in Figure 2.2:

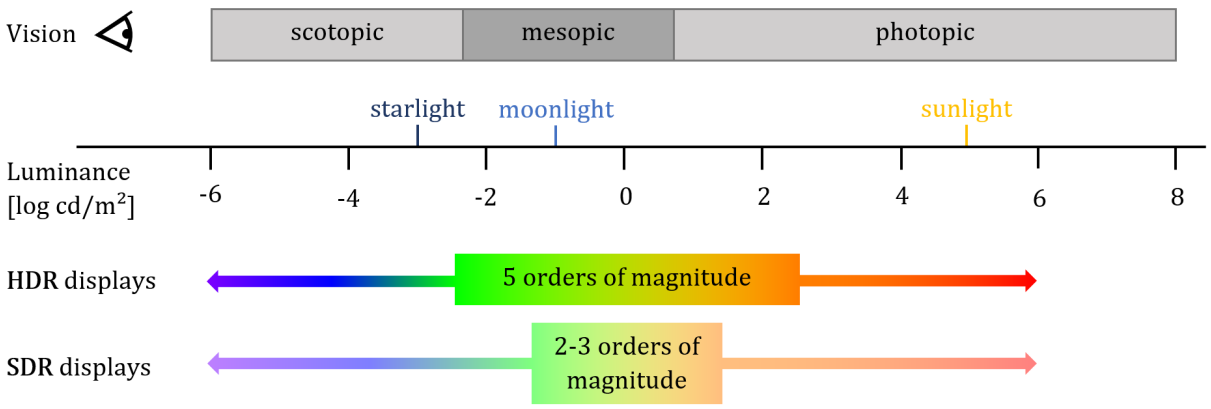


Figure 2.1: Dynamic range of the human vision. *After Hood 1986* [11]

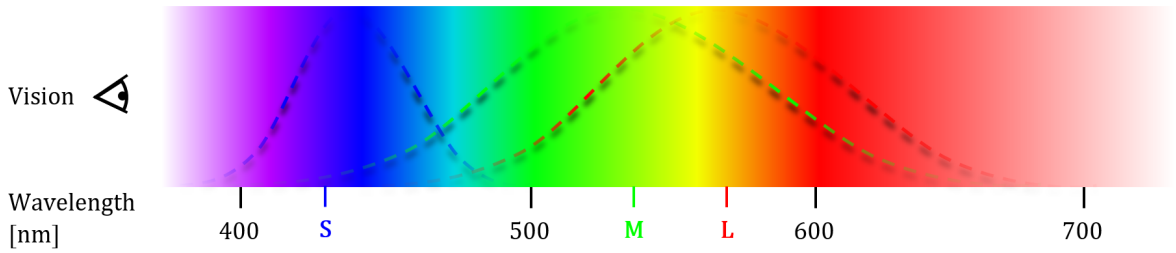


Figure 2.2: Visible color spectrum.

— **S-cones** corresponding to short-wavelength (around 420nm) are responsible for our perception of blue colors
— **M-cones** corresponding to medium-wavelength (around 530nm) are responsible for our perception of green colors
— **L-cones** corresponding to long-wavelength (around 560nm) are responsible for our perception of red colors

2.2 Color spaces

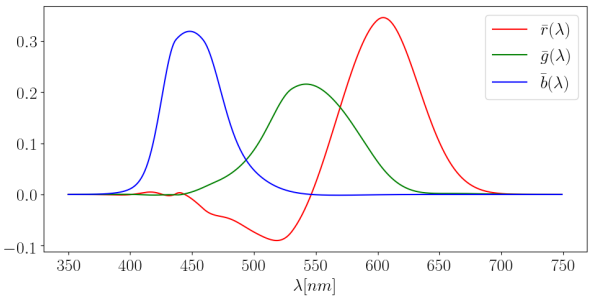
According to the trichromatic theory of color vision [12], each color C of the visible spectrum is equivalent to a linear combination of three primary colors red (R), green (G) and blue (B):

$$C \equiv R + G + B \tag{2.1}$$

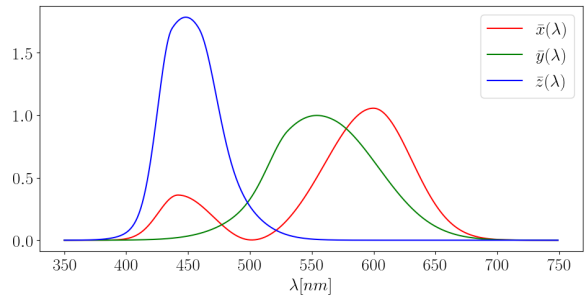
where each component is the response of our three cones $\bar{r}(\lambda)$, $\bar{g}(\lambda)$ and $\bar{b}(\lambda)$ for a given Spectral Power Distribution $S(\lambda)$:

$$\begin{aligned} R &= \int S(\lambda)\bar{r}(\lambda)d\lambda \\ G &= \int S(\lambda)\bar{g}(\lambda)d\lambda \\ B &= \int S(\lambda)\bar{b}(\lambda)d\lambda \end{aligned} \tag{2.2}$$

Finally, experiments on human color perception [13], [14] allowed to define the tristimulus functions $\bar{r}(\lambda)$, $\bar{g}(\lambda)$ and $\bar{b}(\lambda)$. However, to be able to represent every possible color C of the visible spectrum, the defined tristimulus functions may have values lower than zero as illustrated in Figure 2.3a. To overcome this issue the International Commission on Illumination (CIE) proposed the XYZ color space in 1931. This new color space, with the



(a) Tristimulus functions of the CIERGB color space.



(b) Tristimulus functions of the CIEXYZ color space.

Figure 2.3: Tristimulus functions corresponding to (a) CIERGB color space and (b) CIEXYZ color space.

corresponding tristimulus functions showed in Figure 2.3b, is a linear transformation of the previously defined RGB color space. However, these color spaces are strongly correlated, a small modification in any of the three components will widely change the perceived color. To tackle this issue, several color spaces have been proposed to separate the color attributes to the light intensity, like YUV or YCbCr [15]. For these decorrelated color spaces, the first component describes the light intensity while the two last components describe the color. Furthermore, as detailed in the following, other color spaces have been modeled to better fit with the human perception of light intensity and colors (HSL [16], L*a*b* [17], L*C*h, etc.)

2.3 Relative perception

Our perception of the light intensity and the color is not proportionally linear to the physical values and varies depending on the visual conditions. As an example, our perception of the light intensity and color depends on the background as illustrated in Figure 2.4. Color Appearance Models (CAMs) aim to mathematically model the relative perception of the human eye, taking into account the luminance of the background and the relative luminance of the surround for example. We first define some terms related to the color perception according to Fairchild [10]:

— **Luminance:** physical quantity of light emitted by an area in cd/m^2.

— **Brightness:** attribute of a visual sensation according to which an area appears to emit more or less light.

— **Lightness:** the brightness of an area judged relative to the brightness of a similarly

(a) Relative perception of the light intensity. The bar in the center is uniformly grey while it appears brighter in the left than in the right.

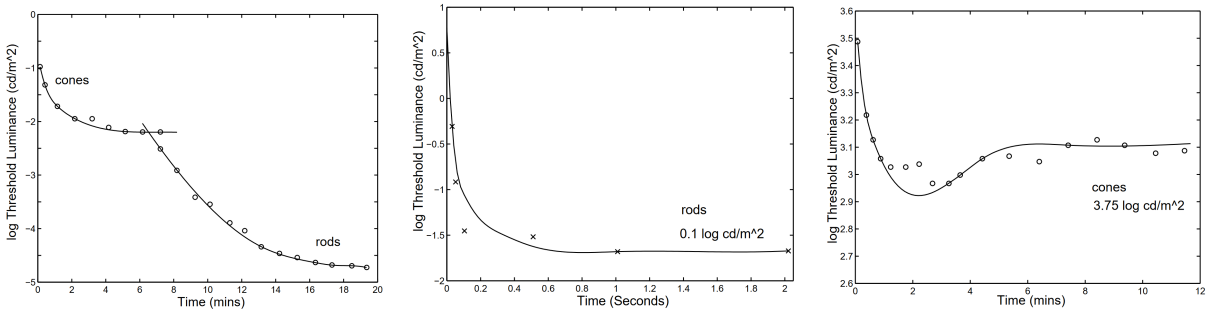
(b) Relative perception of the color. The bar in the center is uniformly yellow while it appears more colored in the left than in the right.

Figure 2.4: Our perception of the light intensity and the color depends on the viewing conditions. In these two illusions, the bars in the center are uniform while we perceive variations because of the gradient backgrounds.

illuminated area that appears to be white or highly transmitting.

- **Colorfulness:** attribute of a visual sensation according to which the perceived color of an area appears to be more or less chromatic.
- **Chroma:** colorfulness of an area judged as a proportion of the brightness of a similarly illuminated area that appears white or highly transmitting.
- **Saturation:** colorfulness of an area judged in proportion to its brightness.
- **Hue:** attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors: red, yellow, green, and blue, or to a combination of two of them.

For example, the L*C*h color space, which stands for the lightness, chroma, and hue respectively, correlates well with how the human eye perceives color. Regarding HDR imaging, many TMOs are based on CAMs to produce pleasant and coherent tone-mapped images. The basis of psychophysical studies that lead to perceptually coherent TMOs is presented in Chapter 5. Moreover, instead of accounting for perceptual coherency, TMOs have to preserve the spatial and temporal coherency of HDR contents. Temporal coherency may be tackled by accounting for the adaptation of the HVS [18] to light changes: from bright to dark and vice versa. While the bright adaptation is fast (few seconds) due to neural processes, the dark adaptation may take up to 20 minutes because of slow chemical processes as shown in Figure 2.5.



(a) The time course of dark adaptation for both rods and cones.

(b) The time course of bright adaptation for rods.

(c) The time course of bright adaptation for cones.

Figure 2.5: Measured time course of bright and dark adaptation for both rods and cones. *Figures courtesy of Ferwerda et al. [18].*

2.4 Conclusion

To sum up, our perception of light and color is complex and strongly depends on the environment of visualization. Conducting experiments on perception for a specific environment of visualization, such as HMD, and modeling the color appearance allows improving the perceived quality of images. Furthermore, having models of light and color perception is even more essential when processing HDR images as detailed in the next chapter.

Chapter 3

High Dynamic Range

High Dynamic Range technology has deeply changed the way of dealing with digital images. From capturing to storing and displaying, the entire imaging pipeline has been transformed [19], [20]. In this chapter, we start by presenting the HDR pipeline. Then, we focus on TMOs, and more specifically on tone mapping dynamic and interactive HDR contents.

3.1 HDR pipeline

3.1.1 Capturing

A straightforward manner of capturing a scene of the real-world is to use an ideal camera that has the same the human eye abilities in terms of dynamic range (see Figure 2.1). For now, HDR cameras that can reach the limits of the human eye perception exist, but are still quite expansive (several thousand dollars). Another way to capture an HDR image is to combine multiple exposures of the same scene captured with a standard camera. While each exposure will have under and overexposed areas due to the limited dynamic range of the camera, zones that show details will be combined to recreate the high dynamic range of the scene (see Figures 1.4 and 1.5). In another way, HDR images may also be synthetically generated by Physically Based Rendering (PBR) in computer graphics.

3.1.2 Storing

Before HDR technologies, each RGB color channel of digital images was encoded on 8 bits, allowing integer values between 0 and 255. Regarding HDR images, the real-world luminance value is encoded using floating-point numbers on the three color channels. The precision of the numerical value depends on the used number of bits, it is generally

between 10 and 16. Thus, new image data format have been developed (like OpenEXR [1] or RGBE [21]) as well as compression algorithms [22]–[25] and new standards (JPEG-HDR, JPEG-XT).

3.1.3 Displaying

Since few years, HDR screens reach the market with the ability to display images with high contrasts (like the SIM2 or the BVM-X300 to cite a few). These new screens tend to reproduce all the dynamic range the human eye can perceive. Consequently, they allowed new research fields around the subjective quality of HDR images [26]–[30]. On the other hand, SDR screens are still used and a way to visualize HDR images on them is to compress the high dynamic range of the content to fit with the limited dynamic range of the screens. This process is ensured by Tone Mapping Operators (TMOs) that convert real-world luminance to display luminance. There exist many different TMOs that produce very different tone-mapped images as detailed in the following section.

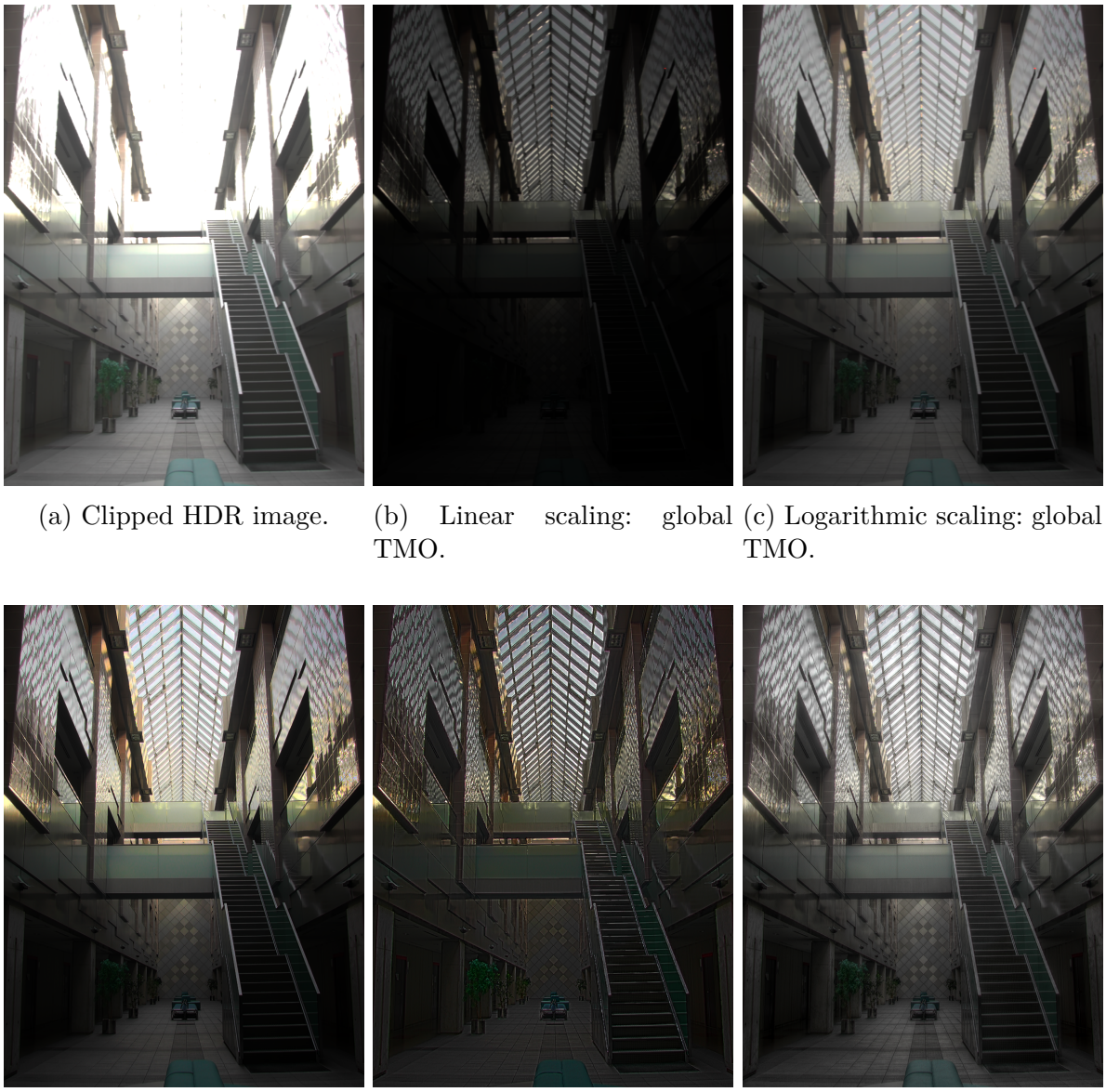
3.2 Tone Mapping Operators

The main goal of TMOs is to display HDR images on SDR screens by preserving as much as possible local and global contrasts while being perceptually coherent. We first detail some TMOs that have been developed for static 2D images before presenting the coherency issues that occur when tackling dynamic 3D contents.

3.2.1 TMOs for still 2D images

We can distinguish two main groups of TMOs: global and local operators. Global operators process all pixels of an image with the same operation [31]–[34] while local ones process each pixel according to its neighborhood [35], [36]. As an example, a simple global TMO is a linear scaling of the luminance values of the HDR image to fit with the limited dynamic range of the display. However, this operation does not account for the human eye perception of the lightness, resulting in a perceptually incoherent image with low contrasts as illustrated in Figure 3.1b. Conversely, applying a scaling in the logarithmic domain produces a perceptually coherent image as shown in Figure 3.1c. The relation between the logarithm of the luminance and our perception of the luminance is detailed

1. `https://www.openexr.com/`



(a) Clipped HDR image.

(b) Linear scaling: global TMO.

(c) Logarithmic scaling: global TMO.

(d) Drago *et al.* [34]: global TMO.

(e) Ashikhmin [36]: local TMO.

(f) Mantiuk *et al.* [37]: combination of global and local TMOs.

Figure 3.1: Comparison between TMOs based on local and global operators. *Images courtesy of Frédéric Drago.*

in Chapter 5. While better image quality may be achieved using more complex global TMOs (see Figure 3.1d), fine details and local contrasts will always be lost. Regarding local TMOs, the operation is different for each pixel, depending on its local neighborhood. The resulting tone-mapped images from local TMOs show a lot of details while the global contrast is lost (see Figure 3.1e). Finally, TMOs that combine local and global operators preserve the global image coherency and enhance fine details and local contrasts as showed in Figure 3.1f.

Otherwise, many TMOs are based on the human eye perception and how it accommodates bright and dark luminances to produce a pleasant and coherent tone mapped image [38], [39]. Other techniques exist like frequency-based [40], [41] or segmentation-based [1], [42] TMOs. Furthermore, instead of preserving the perceptual coherency, TMOs should account for spatial and temporal coherency when processing a sequence of images, such as HDR videos or free walk-through HDR 3D point clouds.

3.2.2 Global coherency for dynamic HDR contents

Spatial and temporal coherency are two aspects of what we call the global coherency of a content. Figure 3.2 illustrates the cases where coherency issues may appear. When working on static 2D images, the notion of coherency only exists in the domain of the image. The temporal coherency happens when the image changes over time, like in videos [43]–[47]. On the other hand, the spatial coherency occurs when visualizing a static scene that has areas with different lighting aspects [7], [8], [48] (like with panoramas or 3D point clouds). In Chapter 6 we tackle the spatial coherency issue when visualizing a 360° HDR image. In a free walk-through interactive 3D content, with a scene composed of manifold areas with dynamic lighting that may change over time (turn on lights or open doors), both spatial and temporal coherency have to be considered [9]. In a free walk-through video HDR 3D point clouds, the conditions are similar. Both spatial and temporal coherency during a free walk-through 3D scenes is the challenge tackled in Chapter 7.

3.3 Conclusion

To conclude, HDR imaging is the upcoming standard by proposing images with more contrasts and more details. However, having efficient TMOs that produce coherent images while preserving fine details is essential for appreciating such HDR images on SDR screens.

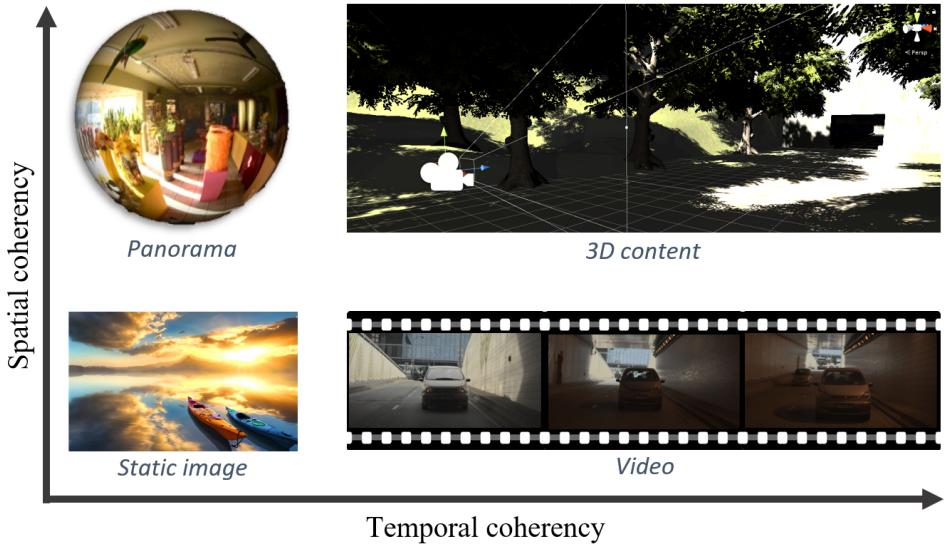


Figure 3.2: Spatial and temporal coherency for tone mapping HDR contents. The notion of coherency is not relevant for static images. The temporal coherency occurs in videos while the spatial coherency occurs in panoramas. Finally, in an interactive 3D content, both spatial and temporal have to be considered.

Our goal in this thesis is not to improve the quality of TMOs for 2D images, but to well define the problem of spatial coherency when tone mapping HDR 3D contents. We propose solutions that make this spatial coherency preserved when visualizing HDR contents on SDR screens and HMDs as explained in Chapters 6 and 7.

Chapter 4

3D POINT CLOUD

While pixels are the primitive elements of images, points are the primitive elements of 3D objects. On one hand, images consist of pixel matrices, on the other hand, 3D objects consist of point clouds. More formally, a point cloud is a set of data points in a 3-dimensional space. A point is generally defined by its position in space (x, y, z) and its color (r, g, b), but it can have many other attributes like the direction of its normal to the surface (n_x, n_y, n_z), a radius r and so on. In this chapter, as in the previous one, we will follow the three steps of the point cloud pipeline [49]: capturing, storing, and rendering.

4.1 Point cloud pipeline

4.1.1 Capturing

There exist several 3D acquisition systems that are used to capture the geometry of the real world. The LiDAR (Light Detection and Ranging) technologies calculate the distance from the sensor to surfaces by measuring the time spent from emitting a light ray to detecting the reflected light ray. These kinds of sensors are frequently used for 3D reconstruction of archaeological sites [2], [3] or urban environments [4], [5] and are also embedded in autonomous vehicles. Another way to capture the distance from a sensor to real-world objects is to use multi-view camera systems and to reconstruct the image's depth [50]. Like humans, our brain estimates the distance of objects by processing two different images (visualized from our two eyes) of the scene that stands in front of us. Thereby, depending on the disparity between the contents of the two images, the depth of the scene can be reconstructed. Those reconstruction techniques often result in RGBD images, which are another representation of point clouds. While the RGB components correspond to the color of the points, the position of the points is given by the pixel coordinates (x, y) and the depth D. However, these 3D acquisition systems assume that the world is fully diffuse involving no specular reflections or refraction, which is a severe

constraint. Otherwise, estimating the depth and other scene attributes from a single image or a video sequence is a current trend for deep learning-based methods [51]–[54]. Once again, point clouds may also be synthetically generated by computer graphics.

4.1.2 Storing

In classical data formats (.ply [55] and .pcd [56] to cite a few), the point clouds are stored as a list of points. Each point consists of a list of attribute values defining its position, color, etc. The question of streaming such point clouds is also discussed, bringing up compression algorithms [57]–[59], quality metrics [60]–[62] and new standards (MPEG-PCC [63], [64]). Instead of saving memory and bit-rate, some works have proposed to arrange the organization of a stored point cloud to speed up the rendering process as detailed later.

4.1.3 Rendering

Point as a primitive

The rendering is the action of displaying a point cloud on a screen, considering its attributes of color, texture, normal, or any other attribute that may alter the visual aspect of the point cloud. A simple way of rendering a point cloud, considering a virtual camera in the scene, is to color the unique pixel that corresponds to the projection of each point. Indeed, if the point cloud is very dense (i.e. a lot of points close together), all pixels of the screen will be colored and the image will be fully rendered. Unfortunately, most point clouds are sparse, which means there is a lot of space between the points. Because points do not cover all the pixels of the screen, the rendered images contain holes. To avoid this issue, points can be represented as surface elements [65], [66] that cover more than one pixel. These surface elements are then splatted onto the screen with a defined radius and with an orientation that may depend on the direction of their normal as illustrated in Figure 4.1. Moreover, assigning an appropriate radius (i.e. the surface covered by the point in number of pixels) and a coherent shape to the points [67], [68] are still open research fields. Nevertheless, holes can still appear in the rendered image and holes filling methods [69]–[71] can be applied. Instead of having a qualitative point cloud rendering, point clouds can be huge (billion of points), taking a while to render. Some interesting structures allow a real-time rendering of huge point clouds as explained in the following.

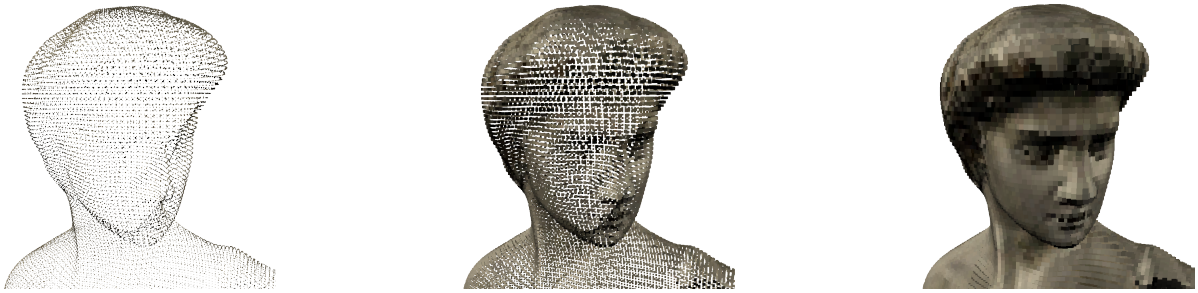


Figure 4.1: From left to right: David point cloud rendered with points radius equal to 1, 3, and 5 pixels respectively. When the radius of the points is large enough, holes are not visible anymore.

Real-time rendering

The first proposed structure that accelerates the rendering of point clouds is named QSplat [72], [73]. Based on a binary tree, QSplat gathers points with their close neighborhoods together in an iterative process. It results in a pyramidal structure that coarsely renders the point cloud at first, and then refines the rendering progressively. Another data structure that allows real-time point cloud rendering, denoted as Potree [74], is based on an octree. Spatially subdividing the scene into a regular grid with several levels of detail, points are gathered by blocks and rendered depending on their visibility and their distance to the camera.

However, the power of modern computers and the evolution of GPU architecture allow rendering billions of points in real-time. As it has been foretold in the pioneer works of Levoy and Whitted [75] about the use of point as a display primitive, digital images tend to be more and more rendered with point clouds. Computer graphics are generalizing the use of point clouds, the number of triangles to render on screen is constantly increasing and the number of pixels each triangle covers is proportionally decreasing. This year, the Epic Games company showed a real-time demo of their Unreal Engine 5 running live on PlayStation 5 [1]. All the assets used for the virtual environment in this demo result from real-world object scans. Therefore, billions of triangles are rendered in real-time with global illumination as showed in Figure 4.2. This demonstration let us think that having a direct rendering of the point cloud of the scanned environment, without an extra process of generating a mesh by triangulation, may be more straightforward and efficient.

1. https://www.unrealengine.com/en-US/blog/a-first-look-at-unreal-engine-5

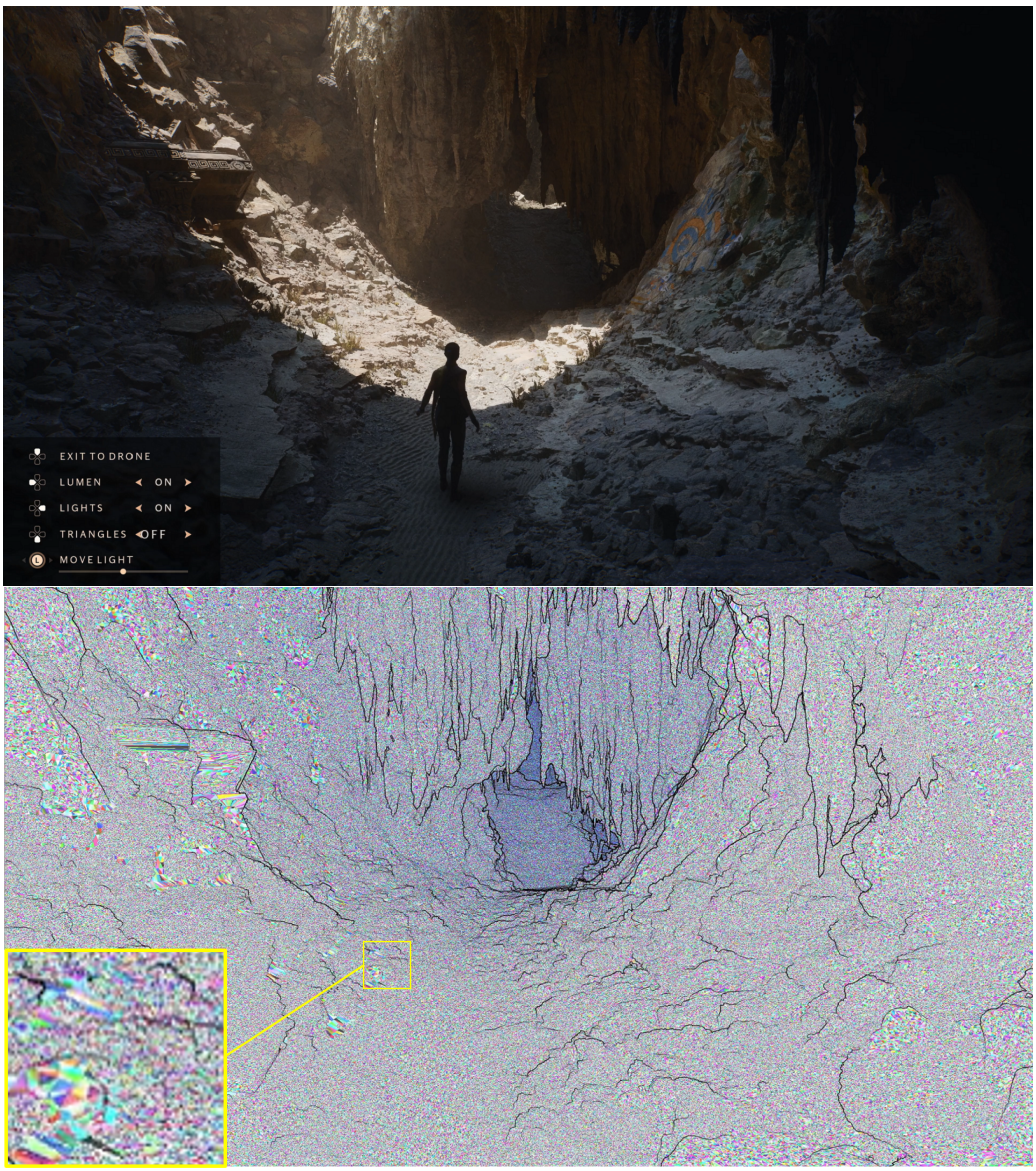


Figure 4.2: Billions of triangles are rendered in real-time with Unreal Engine 5. Each triangle is rendered with a random color, as most triangles cover less than a pixel, the image appears noisy.

4.2 Conclusion

To conclude, a point cloud is a data format that may be rendered efficiently. There is room for improvement regarding the rendering quality of such point clouds, and there is strong interest in this topic. In this thesis, we worked on color transfer between point clouds as detailed in Chapter 8.

Part II

Human eye perception on Head Mounted Display

INTRODUCTION

One challenge in this thesis was to develop a TMO for HDR 3D point clouds visualized on SDR screens and HMDs. The few works about TMO for HDR panorama visualized on HMD showed that existing methods produce a low image quality. We decided to study the human eye perception on such HMD and proposed a new TMO adapted to these specific visualization conditions.

In the following chapter, we present existing CAMs and explain why these models are not adapted to the perception on HMDs. To develop a TMO adapted to the human eye perception on HMDs we conducted two subjective evaluations. We found that the perception on HMD differs from the one on classic 2D display. Finally, we improved an existing CAM by adapting it to the perception on HMD thanks to the measures obtained with our subjective evaluations.

Chapter 5

Lightness and color perception on Head Mounted Display

5.1 Introduction

Due to the growth of VR technologies over the last years, the visualization of 360° images has become common. Moreover, HDR cameras are now used to capture the whole dynamic of a scene with much more details in the brightest and darkest areas, thereby providing realistic panoramas. Nonetheless, the huge majority of manufactured HMDs still have SDR screens, which prevent them from displaying all the dynamic range of HDR images. To appreciate HDR contents through standard displays, the well-known process of tone mapping is used to get a limited range corresponding to that of SDR displays. Many TMOs exist [19], [20] and can be divided into two main groups (global and local) and are often based on how the human perceives lightness and colors. To adapt existing TMOs to HMD visualization, we conducted two subjective evaluations to investigate how the HVS perceives images on HMDs. Section 5.2 introduces related work on perception models. Then, we present the subjective evaluations we conducted to model lightness and color perception in Section 5.3. As a result, we show that the perception model on classic 2D display is slightly different than on HMD.

5.2 Related work

The basis of psychophysical studies about lightness perception comes from the seminal work of Weber. He showed that the human capacity to distinguish a stimulus from a background is linearly proportional to the background luminance. In other words, the lighter is the background L, the higher should be the difference ΔL (between the stimulus and the background) to perceive the stimulus. This ratio is commonly known as the Just

Noticeable Difference (JND):

$$JND = \frac{\Delta L}{L} = k \tag{5.1}$$

with ΔL the luminance difference between the stimulus and the background (in cd/m^2), L the background luminance (in cd/m^2) and k a constant (around 0.01 for traditional visualization condition on a 2D display [19]).

Thereafter, based on the Weber's result, Fechner defined the response of the visual system [76]:

$$\frac{dR}{dl}(L) = \frac{1}{\Delta L(L)} \tag{5.2}$$

where, for a given background luminance L, R is the response of the visual system and ΔL is actually the perceived difference measured in the Weber's experiment. The left part of Equation 5.2 expresses the capacity to perceive a change dR when increasing the luminance intensity dl of a stimulus for a given background luminance L. Written differently, it is equal to the perception of a change (related to the value of 1) when the difference ΔL between the background L and the stimulus is perceptible. Finally, Fechner integrated Equation 5.2 to find the function of the visual system response R for a given background luminance L:

$$R(L) = \int_0^L \frac{1}{\Delta L(l)} dl \tag{5.3}$$

Replacing ΔL by kL from Equation 5.1, we obtain:

$$R(L) = \int_0^L \frac{1}{kL(l)} dl = \frac{1}{k} ln(L) \tag{5.4}$$

Accordingly, the response of the visual system R for a given luminance L is equal to the logarithm of this luminance, up to a constant. The response of the visual system is called the lightness, the subjective perception of the luminance of a stimulus according to the viewing conditions, as defined in [10].

More recently, Stevens showed limits of the Fechner's model and proposed to use a power function to model the lightness perception [77]. Stevens' psychophysical studies have led to the lightness perception equal to the physical luminance raised to the power of one third (the lightness component in the CIE L*a*b* color space is based on this result). Both of those models are still used and seem to give similar results in comparable conditions (see Figure 5.1). The debate to know which representation of the lightness perception is more accurate is still open and research continues on this topic [78].

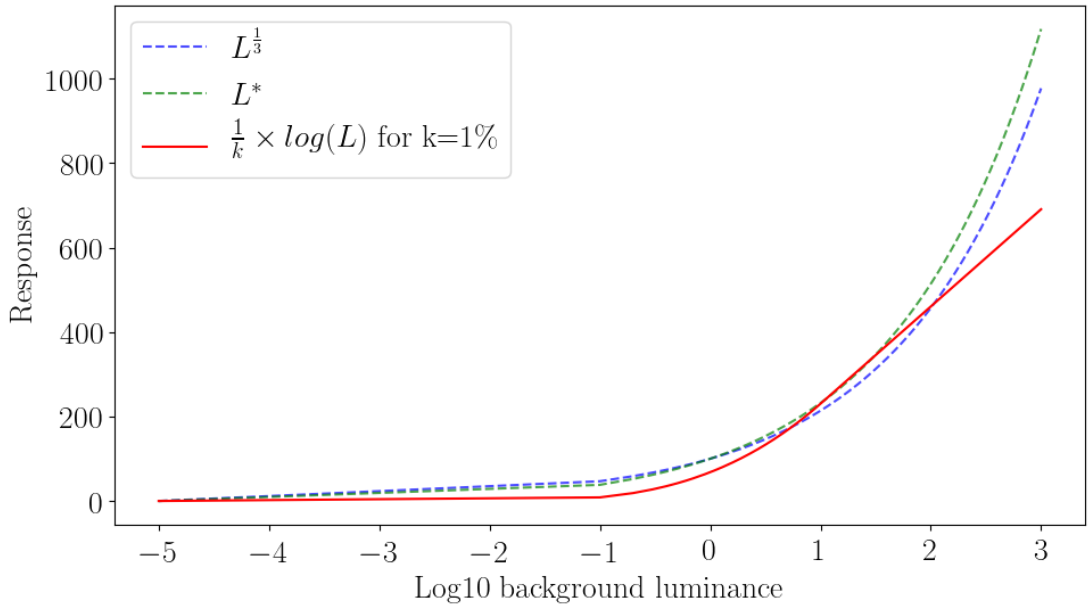


Figure 5.1: Lightness functions comparison.

Decades later, Whittle conducted a subjective evaluation on lightness perception following a different protocol [79]. He measured the JND between two stimuli (respectively the reference and the test) in front of a unique background. In his experience, the reference stimulus and the background have fixed luminances (the reference stimulus can be brighter or darker than the background) while the test stimulus is varying incrementally until the participant perceives a difference between the two stimuli. The obtained results are similar to Weber's ones, the luminance discrimination is equal to a constant k:

$$\frac{|W_t - W_r|}{W_r} = k, \qquad \text{with } W_t = \frac{\Delta L_t}{L_{min}} \text{ and } W_r = \frac{\Delta L_r}{L_{min}} \tag{5.5}$$

where ΔL_t and ΔL_r are the luminance difference between the background and the test or the reference stimulus respectively and L_{min} is the minimum luminance value between the reference stimulus and the background. This kind of representation is sufficient as a simple model of lightness perception, but current CAMs consider more parameters of the HVS, such as the chrominance and the viewing conditions.

In the CIECAM02 proposed by Fairchild [10] (recently improved by the CAM16 by Li *et al.* [80]), the perceived lightness J of a stimulus depends on the luminance stimulus,

the luminance background, and the lighting conditions of the surround:

$$J = 100 \times \left(\frac{A}{A_w}\right)^{c.z}, \qquad \text{with } z = 1.48 + \sqrt{\frac{Y_b}{Y_w}} \tag{5.6}$$

where A and A_w are the achromatic response of the stimulus and the achromatic response of the white reference respectively. z corresponds to the base exponential nonlinearity with Y_b and Y_w the background luminance and the white reference luminance respectively. Moreover, this model considers the surround enlightening (see Figure 5.3a) as it influences the lightness perception. Finally, the value of the nonlinearity factor of brightness c is defined depending on the viewing condition (the surround) that can be Dark, Diminish or Average, as illustrated in Figure 5.2.

On an HMD, the viewing conditions are not well defined. As described later in Section 5.3, we suppose that the surround component is a function of the background, and the background luminance influences much more the lightness perception than on classic 2D display. The influence of the size of the background and a complex enlightening environment on 2D displays has been already studied in [81]. To obtain a nonlinearity factor of brightness c that directly depends on the background luminance, Lee and Kyu-Ik [82] proposed some adjustments in the model. First, an adaptation degree F is expressed depending on the background luminance L_b instead of the three original viewing conditions

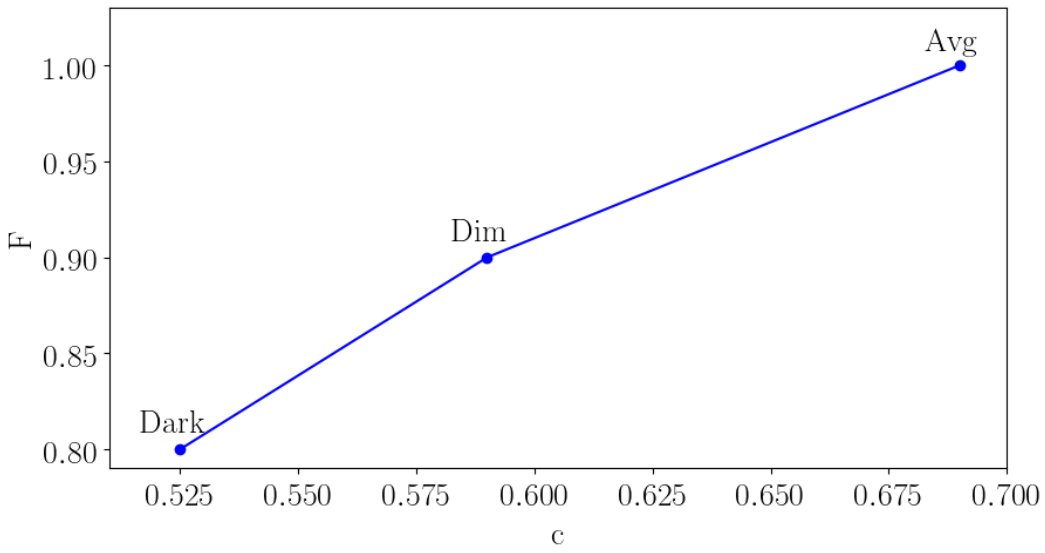


Figure 5.2: Maximum degree of adaptation F depending on the surround component c related to the viewing condition (Dark, Diminish or Average).

(Dark, Dim, Avg):

$$F = \begin{cases} 0.7379 + 0.392\ \Big(1 - exp(0.0221 \times L_b)\Big), & \text{if } L_b < 50\ cd/m^2 \\ 1, & \text{otherwise} \end{cases} \tag{5.7}$$

where the adaptation degree F is only a function of the background luminance. Then, the new nonlinearity factor, called now c_L, is computed depending on an adaptation luminance L_a:

$$L_a = F.L_b + 0.2(1 - F)L_{d_{max}} \tag{5.8}$$

where $L_{d_{max}}$ is the maximum luminance of the display and L_a is the luminance of adaptation (20% of $L_{d_{max}}$ in traditional viewing condition). Finally, the adapted nonlinearity factor c_L is calculated as follows:

$$c_L = \frac{c.\Delta L_{a|L_a=50}}{\Delta L_a}, \qquad \text{with } \Delta L_a = 1.88L_a^{0.23} - 7.24L_a^{0.11} + 8.26 \tag{5.9}$$

where the nonlinearity factor of brightness c is equal to 0.69. $L_a = 50$ is a limit defined by the authors where $\Delta L_{a|L_a=50}$ is equal to 1.75.

This complex model allows us to express the nonlinearity adaptation factor c_L and the adaptation degree F as a function of the background luminance. Nevertheless, the nonlinearity factor c_L (equation 5.9) has to be adapted to fit with the particular viewing conditions encountered on HMDs. We ran subjective evaluations to compare perception on HMD with known models and found that the lightness perception is halved on an HMD compared with a 2D display (see Section 5.3).

5.3 Perception on Head Mounted Display

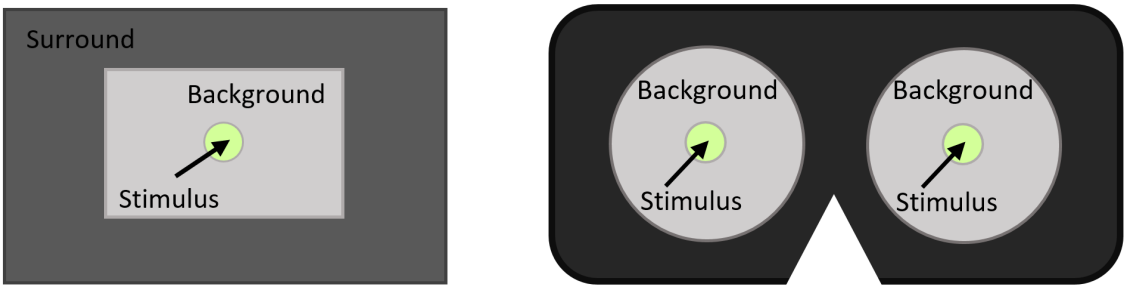
Before delving into the proposed TMO dedicated to HMD, we describe two subjective experiments we conducted to study the HVS response on HMD. The first one focuses on lightness perception and measures the JND in the dynamic range of the HMD. Moreover, the protocol design follows CIECAM recommendations [10]. This experiment is helpful to validate the use of the logarithm of the luminance as a good representation of the perceived lightness. The second subjective evaluation is intended to be more general, regarding lightness and saturation. The experiment is based on the Whittle design [79], consisting of presenting two stimuli in front of a uniform background. Its usefulness is

twofold. First, it confirms the result of the previous experiment about lightness. Second, it allows an evaluation of the perception of the chrominance on HMD. Resulting from these two evaluations, we propose to adapt a CAM to make it coherent with the perception on HMDs. Our CAM, called HMDCAM from now on, is used in our TMO as described later in Chapter 6.

5.3.1 First experiment: lightness as a function of luminance

The classic Weber's experiment seeks to determine the minimum perceptible difference value of luminance between a stimulus and a background. At first, a fixed background is presented to participants with the stimulus of the same luminance. Then, the stimulus luminance is increased until participants notice a difference between the stimulus and the background. This relative difference is called the Just Noticeable Difference. Moreover, as the perception depends on the environment enlightening, CIECAM [10] suggests considering the surround luminance as a parameter. CIECAM recommends the stimulus to have a radius ranging from 2° to 4° in the visual field (corresponding to the foveal vision), a radius of 20° for the background (peripheral vision) and the surround encompasses the rest of the field of view (see Figure 5.3a).

These recommendations are well defined in the case of visualization on a classic 2D display. New constraints are met when considering visualization on HMD. First, the surround field is not considered anymore because the black plastic structure of the HMD encompasses the whole visual field. Second, while the CIECAM model has been designed for distant display visualization that covers the vision of about 20°, on an HMD this angle corresponds to all of the Field of View (FoV) of the display (about 100°). To sum up, in



(a) (b)

Figure 5.3: (a) CIECAM02 recommendations for visualization conditions on 2D display. (b) Visualization conditions on HMD.

our experiment on HMD, we consider a 4° stimulus, the background covers all the FoV of the HMD (100°), and the surround field is ignored (see Figure 5.3b). The JND has been determined for ten background luminance levels covering all the dynamic range of the HMD. The test lasted about 15 minutes and the panel consisted of 20 participants (13 men and 7 women) with normal vision, from 20 to 57 years of age, with various socio-cultural backgrounds. After data fitting using robust estimators and classical regressions, we found a linear JND with a slope equal to 2.2% (± 0.3%) as illustrated in Figure 5.4. The sensitivity is still linear (ΔL as a function of L), resulting in a logarithmic response when using the Fechner's integration (Equation 5.4). This evaluation emphasizes that the logarithmic lightness function is still valid to model the human perception on HMD.

However, this JND approximated to 2% is interpreted as a loss of contrast (two times less) for visualization on HMD compared with visualization on classic 2D display where the JND is usually around 1% (see Figure 5.1). We suppose this phenomenon is due to the lack of a fixed luminance value in the surround. CAM proposed by Fairchild [10] considers three viewing conditions for the surround: Dark, Dim, and Average. Indeed, the light emitted from the displays in the headset scatters on the plastic structure. Assuming the structure of the HMD is equivalent to the surround field in the CIECAM recommendations, the surround S is then a function of the background B and becomes lighter when

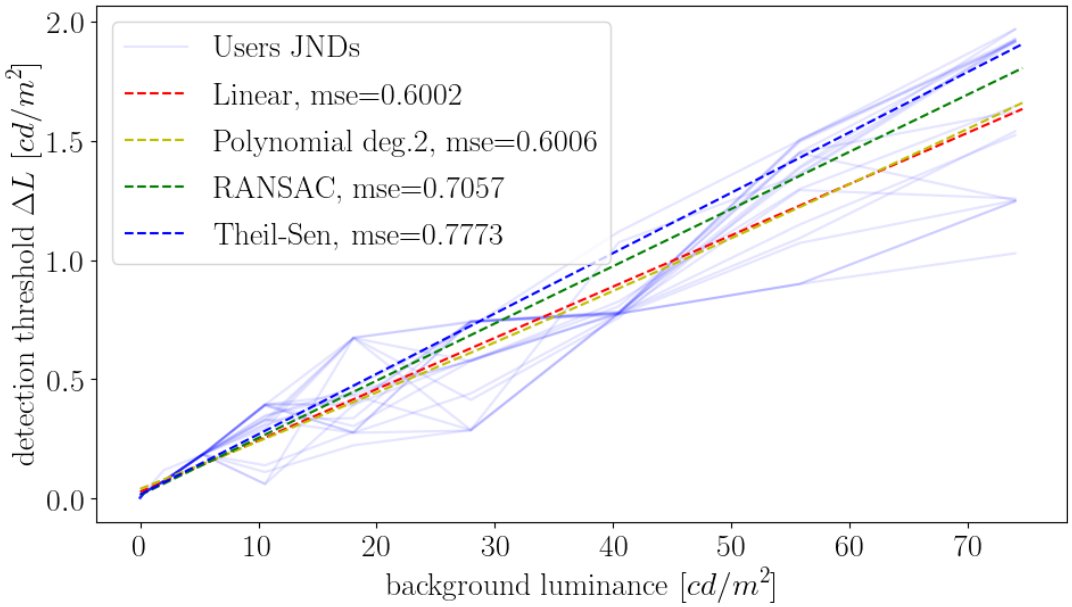


Figure 5.4: ΔL as a function of L given the JND on an HMD.

the luminance of the background increases:

$$S = f(B) \tag{5.10}$$

Based on Lee and Kyu-Ik [82] work that takes into account the influence of background luminance in the lightness equations, we adjusted the viewing-dependent component c_L to fit with our results.

$$c_L = \frac{c.r.\Delta L_{a|L_a=50}}{\Delta L_a}, \qquad \text{with } r = \frac{0.01}{k} \tag{5.11}$$

where r is the ratio between the classic viewing condition constant at 1% and the found constant $k = 2.2\%$ on HMD. c is still equal to 0.69. We scaled the parameter c_L depending on the perception on HMD, which is almost halved. Finally, our lightness model relies on:

$$J = 100 \times \left(\frac{A}{A_w}\right)^{c_L.z}, \qquad \text{with } z = 1.48 + \sqrt{\frac{Y_b}{Y_w}} \tag{5.12}$$

To compare the visual response of our HMDCAM with the CIECAM02, we simulated the difference in lightness perception between two stimuli for a solid background, as illustrated in Figure 5.5. The perceived lightness is strongly attenuated by the background luminance in the case of HMD visualization compared with the three viewing conditions

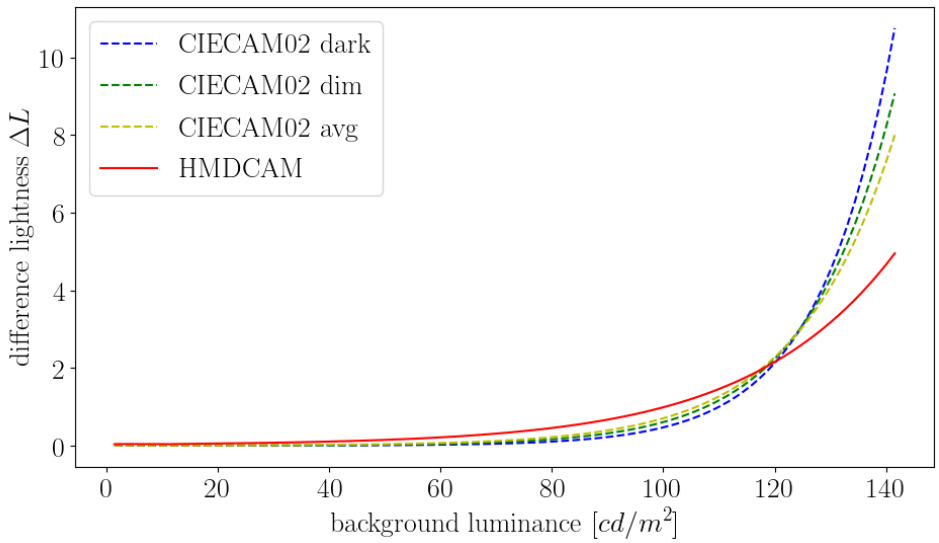


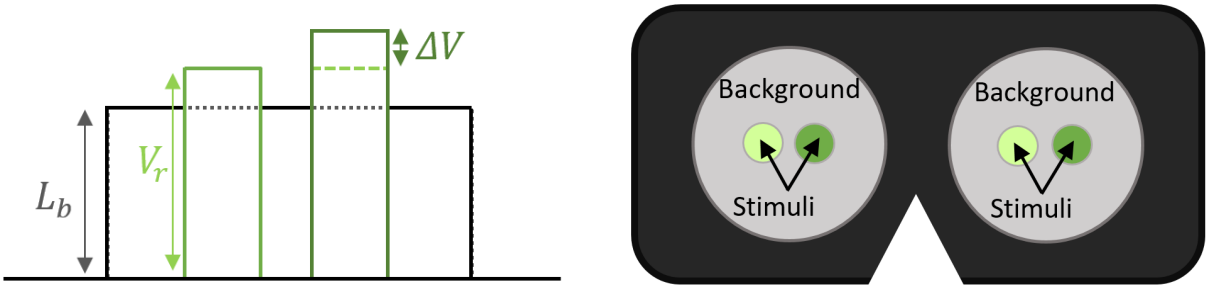
Figure 5.5: Proposed HMDCAM compared with the three viewing conditions of CIECAM02.

proposed in CIECAM02. These two factors (F and c_L Equations 5.7 and 5.11 respectively) are used in the next evaluation and confirm the validity of our lightness model on HMD.

5.3.2 Second experiment: chrominance response function

To validate our HMDCAM proposed above, we conducted a second experiment that includes the evaluation of the chrominance perception. Its design is inspired by Whittle's experiment on luminance discrimination [79]. Instead of measuring the difference between a stimulus and a background, we compare two stimuli over a uniform grey background. We adapted this evaluation to determine JNDs of luminance and saturation for a set of different stimuli (see Figure 5.6). The background L_b is an achromatic luminance in the range of the display. V_r is the chromatic or achromatic reference stimulus. Finally, ΔV is the difference between the reference and the test stimulus. ΔV can be either a difference in luminance L or in chroma C.

In this study, both the background and the reference stimuli are fixed while the test stimulus increases in luminance or saturation until participants notice a difference between the two stimuli. The two stimuli have a radius of 2° each and are separated by 8° of the vision field. The background still covers the entire FoV of the HMD (100°). The panel consisted of 18 participants, from 20 to 57 years of age, with normal vision and various socio-cultural backgrounds. The test lasted about 45 minutes and was split into two parts. First, 8 luminance values are evaluated for 8 different backgrounds, resulting in 32 discrimination luminance values. Then, the saturation has been evaluated for a unique background equal to 20% of the maximum luminance of the HMD (about 30 cd/m^2). The



(a) (b)

Figure 5.6: (a) Schematic experiment with the background luminance L_b, the reference stimulus V_r and the difference ΔV between the two stimuli. (b) Schematic experiment represented on HMD.

second part of the experiment consists of 8 saturation values for 4 colors: red, yellow, green, and blue (32 values in total). One goal of this experiment was to confirm the HMDCAM that has been proposed after the first subjective study. To this end, for the evaluation of the lightness, we compute the CIECAM02 lightness response by averaging the values provided by all the participants for both J_{ref} and J_{test}, the reference and the test stimuli respectively.

$$J_{ref} = 100\Big(\frac{A_{ref}}{A_w}\Big)^{c.z} \qquad \text{and} \qquad J_{test} = 100\Big(\frac{A_{test}}{A_w}\Big)^{c.z} \tag{5.13}$$

where A_{ref} and A_{test} are the achromatic response of the reference and the test stimulus respectively. $c = 0.69$ according to the Average surround lighting condition (as it is the condition with the closest estimation of the lightness perception to our HMDCAM, as showed in Figure 5.5). Then, we compute the absolute difference between the two lightness responses:

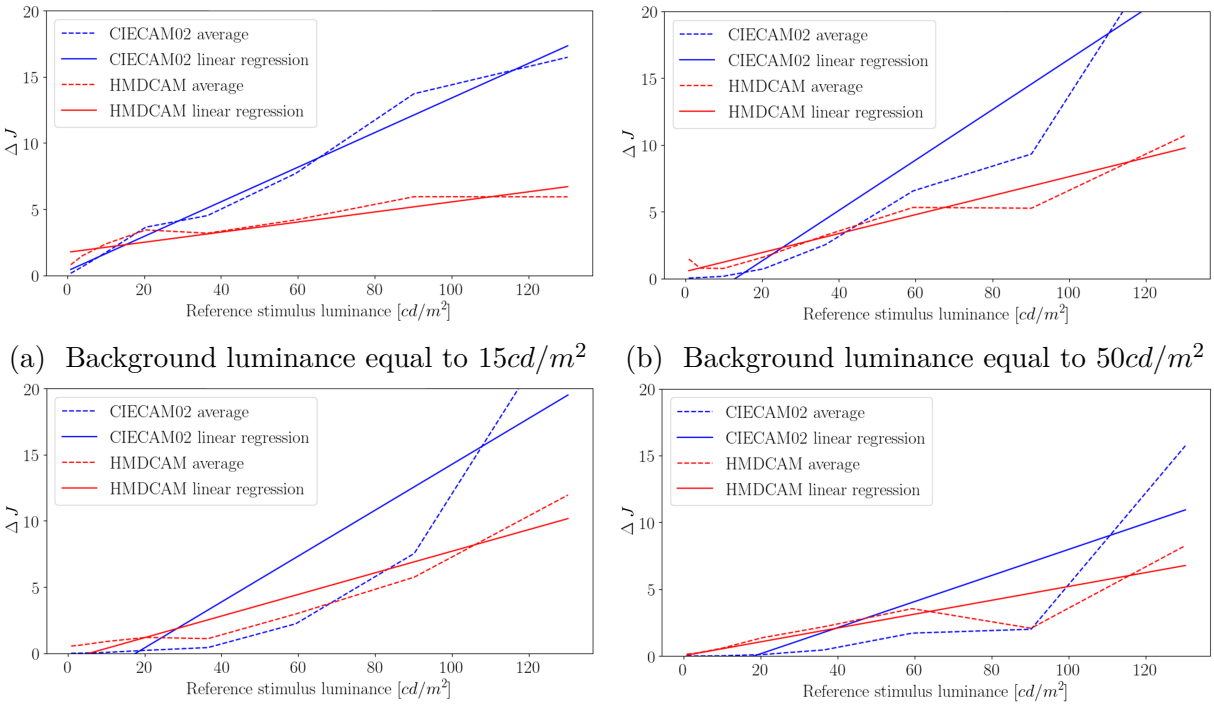
$$\Delta J = |J_{ref} - J_{test}| \tag{5.14}$$

As the test value has been determined as the JND between itself and the reference stimulus, ΔJ should be almost constant for any of the 32 conditions.

We computed ΔJ for all the tested values (in case of lightness evaluation) using either CIECAM02 or HMDCAM and displayed resulting curves on Figure 5.7 for different levels of background luminance.

As a reminder, we expect ΔJ to be constant whatever the background and the reference stimulus luminance. A linear regression over the two curves in each of the four background luminance conditions showed that HMDCAM (red lines) has a lower slope (i.e. is more constant) than CIECAM02 (blue lines). With both models, the more the stimulus luminance increases, the more the difference ΔJ of perceived lightness is significant. It means that HMDCAM can still be improved, even though the ΔJ approaches a constant value (about 6% error on average). The error of estimated lightness for both CIECAM02 and HMDCAM have been computed for all the tested conditions as showed in Table 5.1. Our model improves the estimation of lightness perception on HMD.

We reproduced the same protocol for the evaluation of perceived saturation by computing the absolute difference between the perceived saturation of the reference stimulus and the test stimulus. The proposed HMDCAM slightly improves the results of the perception of saturation (see Table 5.2). Nevertheless, the error does not differ too much between classic viewing conditions and visualization on an HMD because the error of



(a) Background luminance equal to $15cd/m^2$

(b) Background luminance equal to $50cd/m^2$

(c) Background luminance equal to $90cd/m^2$

(d) Background luminance equal to $125cd/m^2$

Figure 5.7: CIECAM02 compared with our HMDCAM model for different levels of background luminance.

Background luminance $[cd/m^2]$	15	50	90	125
CIECAM02 error [%]	13.1	18.8	17.3	9.7
HMDCAM error [%]	**3.8**	**7.1**	**8.2**	**5.2**

Table 5.1: Error of estimated lightness perception (CIECAM02 [10] compared with our proposed HMDCAM).

Color	Red	Green	Blue	Yellow
CIECAM02 error [%]	1.3	0.6	3.2	5.3
HMDCAM error [%]	**0.7**	**0.5**	**1.7**	**2.8**

Table 5.2: Error of estimated saturation perception (CIECAM02 [10] compared with our proposed HMDCAM).

CIECAM02 stays low.

5.4 Conclusion

To sum up, we have seen that the perception on HMD differs from classical perception on 2D display. Our two experiments showed that the perception of a difference between two levels of luminance is halved. We then proposed an HMDCAM that better describes the perception on HMD. In the following chapter, we present our TMO that uses the HMDCAM to improve the quality of the tone-mapped images visualized on HMD.

Part III

Tone mapping High Dynamic Range 3D point cloud

INTRODUCTION

In this part, we will describe the two TMOs we developed for visualizing HDR panoramas and HDR rendering of 3D scenes on HMDs. Our first TMO uses HMDCAM (defined in Chapter 5) to improve the visual quality of tone-mapped images and also considers the whole panorama to preserve the spatial coherency of the scene as detailed in Chapter 6. Then, our second TMO presented in Chapter 7 considers a whole 3D scene to preserve spatial and temporal coherency when moving the camera.

Chapter 6

A perceptually coherent Tone Mapping Operator for Head Mounted Display

6.1 Introduction

In this chapter, we present a new TMO dedicated to the visualization of 360° HDR images on HMDs. Previous works around this topic have shown that existing TMOs for classic 2D images are not adapted to 360° HDR images. Different approaches to address the problem of 360° image tone mapping on HMD can be considered. One solution is to apply the TMO to the whole 360° image, considering its entire dynamic range. The obtained result is globally coherent but, when considering only the part of the image currently viewed by the user (i.e. the viewport), the contrast can be unpleasantly reduced. On the other hand, as the user can only watch a limited part of the 360° image at a time, a TMO may be applied to the current viewport. Thus, the viewport contrast is enhanced while the global coherency is lost. To overcome this problem, we propose a method that takes into account the results of two TMOs: one applied to the entire 360° image, and the other to the current viewport. As explained later, the viewport TMO provides a better contrast while the global TMO preserves the global coherency. Our contribution can be considered as an improved TMO for 360° HDR images that ensures a spatial coherency and enhances contrasts while being perceptually coherent with the lightness perception of the human eye on HMD.

The chapter is organized as follows. Section 6.2 introduces related work about TMOs dedicated to 360° images visualization on HMD. Then, we detail our HMD-TMO in Section 6.3. In Section 6.4, we comment on our results and discuss the efficiency of our approach. Finally, Section 6.5 concludes this chapter and presents some research avenues for future work.

6.2 Related work

Assuming that perception is the same for HMDs and 2D displays, two user studies performed a subjective comparison of several TMOs applied to many 360° HDR images to find the most appropriated TMO for HMD. The first evaluation conducted by Perrin *et al.* [83] consists of applying existing TMOs to the entire 360° HDR image and display the obtained result on the HMD. However, none of the evaluated TMOs shows a clear improvement of perceived quality. The year after, Melo *et al.* [84] conducted another user study to compare four TMOs on five 360° HDR images and found similar results. These results suggest that existing TMOs should be adapted to meet the requirements of 360° HDR images displayed on HMDs.

Additionally, a few TMOs dedicated to HMD have been proposed. Yu [48] adapted the Photographic Tone Reproduction operator [85] to each viewport of a 360° image. For the first time, Yu proposed to apply the TMO to the viewport only, rather than to the entire 360° image. Moreover, as the dynamic range in the viewport can significantly change from a view to another, applying a TMO to each view independently may produce flickering artifacts. To prevent such flickering, Yu proposed to smooth the TMO parameters between successive views, thereby reproducing light and dark adaption of the human vision.

Similarly, Cutchin and Li [86] developed a method that applies a tone mapping to each viewport independently, depending on its luminance histogram. The viewport histograms are divided into four groups corresponding to four different TMOs. The authors noticed popping effects that happen when two successive views belong to different groups.

At last, both methods benefit from view dependency on HMD and provide a better perceptible quality in the viewport. However, they still present some limits we want to overcome. Especially, these two methods do not tackle the spatial coherency as the TMO is applied to the viewport only. We propose a method that takes advantage of the viewport-dependent operation with smooth transitions between successive views to ensure a good contrast while maintaining a global coherency considering the luminance of the entire 360° image. In the next section, we present our perceptually coherent TMO for HMD.

6.3 A new Tone Mapping Operator for Head Mounted Display

Applying a TMO to an entire 360° image (i.e. considering all the dynamic range of the scene) produces a globally coherent tone-mapped result, while the contrast in the viewports is reduced. As seen in previous evaluations [83], [84], applying a TMO to the entire 360° image does not produce a satisfying quality. On the other hand, applying a TMO to a viewport only preserves the contrast but loses the global coherency of the scene. Since none of these methods produces a satisfying result when applied independently, we developed a TMO that combines both methods, global and viewport based, adapted to visualization on HMD to preserve global coherency and enhance contrast.

The framework of our TMO is presented in Figure 6.1 and consists of a pipeline with two branches. The input is a 360° HDR image and the output is a tone mapped-image of the current viewport. The upper branch performs a tone mapping on the entire 360° image and thus preserves the spatial coherency. This operator is based on the log-luminance histogram of the image. We will see that computing a naive histogram of the equirectangular projection of the 360° image leads to an unrepresentative distribution of the luminance. Concurrently, the lower branch performs a tone mapping on the viewport image to enhance the contrast. This operator is based on the Photographic Tone Map-

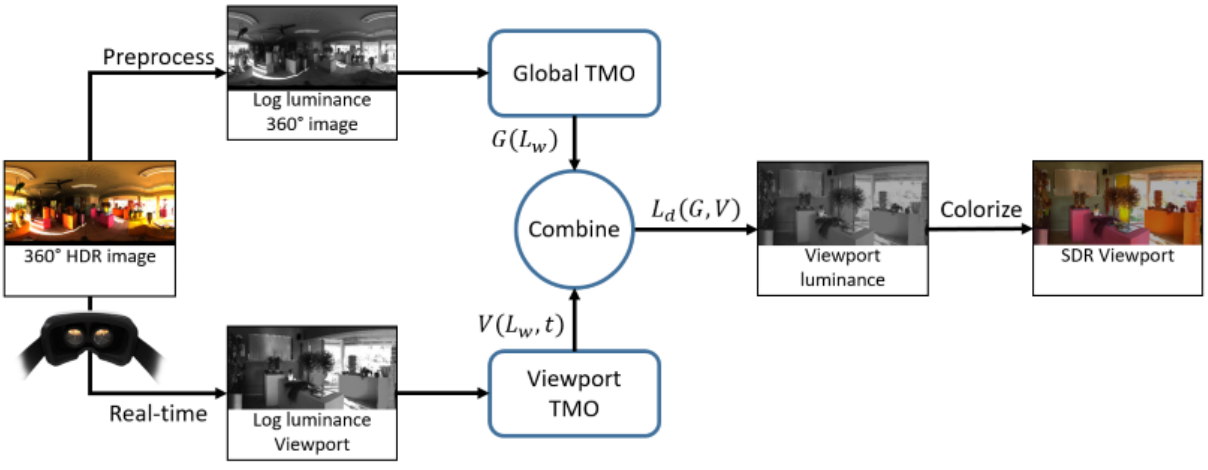


Figure 6.1: Our operator combines a Global TMO $G(L_w)$ and a Viewport TMO $V(L_w, t)$. The Global TMO (upper branch) preserves the global coherency of the scene while the Viewport TMO (lower branch) enhances contrast. The combination of both produces our final HMD-TMO $L_d(G, V)$.

ping Operator [85]. Note that the time parameter t means the viewports succession due to the movement of the user who is wearing the HMD. The combination of the resulting luminances of these two TMOs (tone mapping of the entire 360° image $G(L_w)$ and tone mapping of the viewport image $V(L_w, t)$) is calculated by a weighted sum in the logarithmic domain to compute the final tone-mapped luminance $L_d(G, V)$. Note that the luminance L_w stands for the Y channel in the CIEXYZ color space following the Rec. UIT-R BT.709 standard [87]:

$$L_w = 0.2126 \times R + 0.7152 \times G + 0.0722 \times B \tag{6.1}$$

Each component of our pipeline (global TMO, viewport TMO and combination) are detailed in the following subsections.

6.3.1 Global Tone Mapping Operator

The global TMO is based on the Visibility Matching Tone Reproduction Operator proposed by Ward *et al.* [33]. It consists of a log-luminance histogram equalization scaled into the dynamic range of the display. To avoid artifacts due to too high contrasts in the tone mapped image, the authors add a pass of histogram adjustment that matches with the HVS luminance response. We adapted this step using our HMDCAM (detailed in Chapter 5).

Thus, the log-luminance distribution is needed to compute the TMO, but a naive histogram of the equirectangular projection of the 360° image results in a wrong distribution. Actually, the projection gives more significance to the poles (top and bottom) of the 360° image than to the equatorial area as illustrated by Figure 6.2. To avoid this over-represented contribution in the histogram, we apply a weight to the pixels depending on the elevation in the equirectangular image to obtain a correct distribution [88], [89]:

$$w_{x,y} = cos\left(\pi \times \left(\frac{y}{H} - 0.5\right)\right) \tag{6.2}$$

where $w_{x,y}$ is the weight of the pixel (x, y) (instead of 1), and H is the image height in number of pixels. The histogram is computed in floating numbers, it is then cumulative and normalized to obtain the Cumulative Distribution Function (CDF). This correction is especially needed given that, in general cases, the pod of the 360° camera that captures the HDR image lets a black area in the bottom, which produces an offset in black level

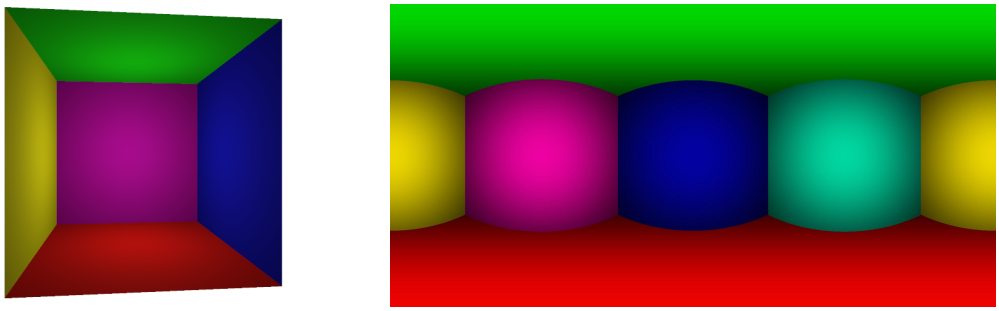
as illustrated in Figure 6.3. Finally, the weighted log-luminance CDF is given by:

$$P(b) = \frac{\sum_{b_i<b} f(b_i)}{\sum_{b_i} f(b_i)}, \qquad \text{with } f(b_i) = \sum_{x,y} w_{x,y} \times log\Big(L_w(x,y)\Big) \tag{6.3}$$

where $f(b_i)$ is the log-luminance weighted sum of all pixels (x, y) that fall into bin b_i. The number of bins is at least equal to 100 to avoid banding artifacts due to quantization. The tone curve G proposed by Ward *et al.* is then a scaled version of $P(b)$:

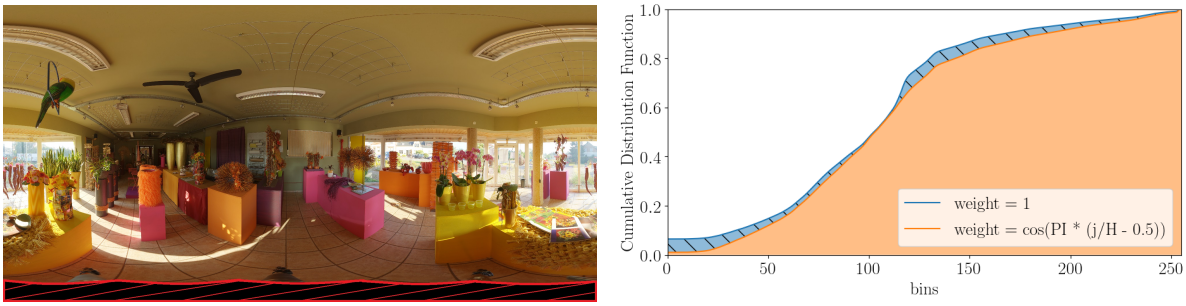
$$G(x,y) = exp\Big(ln(L_{dmin}) + \Big(ln(L_{dmax}) - ln(L_{dmin})\Big) \times P\Big(L_w(x,y)\Big)\Big) \tag{6.4}$$

where L_{dmin} and L_{dmax} are respectively the minimum and maximum luminance of the display and $L_w(x, y)$ is the world luminance of the pixel (x, y). $P(L_w(x, y))$ is the CDF



(a) (b)

Figure 6.2: (a) Inside of a cube, each wall has the same dimension. (b) 360° image (equirectangular projection), the poles (green and red areas) cover half of the image.



(a) Equirectangular projection of Florist 360° image. The camera pod (hatched in red) is over-represented.

(b) Comparison of CDF without weights (blue curve), and with weights (orange curve).

Figure 6.3: The camera pod -hatched in red in (a)- that captures the HDR image is over-represented, producing an offset in black level when calculating the CDF.

defined in Equation 6.3, and $G(x, y)$ is the resulting luminance of the pixel.

At this stage, $G(x, y)$ expresses a tone-mapped luminance only based on the log-luminance histogram of the image. Hence, if the slope of the CDF is too steep, the contrast produced by the tone mapping is too high to fit with the human eye perception. To preserve a perceptually coherent contrast in the image, Ward *et al.* [33] proposed to adjust the histogram based on the human perception. This process is called the histogram ceiling. If the contrast between two levels of displayed luminance is perceptually higher than the contrast between the corresponding levels of world luminance, then the histogram has to be adjusted. It can be written thus:

$$\frac{dL_d}{dL_w} \leq \frac{J_d}{J_w} \tag{6.5}$$

where L_d and L_w are the display luminance and the world luminance respectively while J_d and J_w are the corresponding perceived lightness. With this technique, the resulting tone-mapped image essentially depends on the model used to compute the perceived lightness J. While Ward *et al.* [33] used the detection threshold function proposed by Ferwerda *et al.* [18] to calculate J, we exploit our HMDCAM detailed in Chapter 5 with J defined in Equation 5.12. Note that the histogram ceiling follows the same iterative process proposed by Ward *et al.* [33]. A comparison of the resulting tone mapped images depending on the used lightness model is presented in Figure 6.4. We remind that the human eye is less sensitive on HMD than on classic display (about halved sensitive as seen in Section 5.3). While the CDF without any ceiling (Figure 6.4a) produces too high contrasts, the histogram ceiling with a classic perception model (Figure 6.4c) flattens the image. Finally, the ceiling based on our HMDCAM (Figure 6.4b) better preserves contrast and stays perceptually coherent when visualized on HMD. The contrast is highlighted by the cobblestones in the images.

To conclude, the global TMO preserves the coherency of the scene and is perceptually coherent when visualized on HMD. However, as it considers the entire dynamic range of the 360° image, the contrast in the viewport may be reduced. To enhance the image contrast, a second TMO is applied to the viewport as explained in the following section.

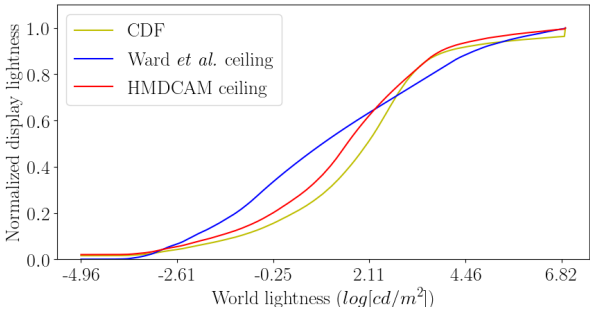
6.3.2 Viewport Tone Mapping Operator

As proposed by Yu [48], our viewport TMO relies on the Photographic Tone Reproduction operator [85] with temporally smoothed parameters to avoid flickering and simulate

(a) Without ceiling: the contrast is too high.

(b) With HMDCAM ceiling: the contrast is coherent with the perception on HMD.

(c) With Ward ceiling: the contrast is too low.



(d) Comparison of the tone curves. Yellow, red and blue curves correspond to (a), (b) and (c) respectively.

Figure 6.4: Histogram adjustment tone mapping using different ceiling.

eye adaptation. This TMO is based on the log-luminance average of the image, called the key value:

$$\bar{L}_w\Big(V(t)\Big) = \frac{1}{N} exp\left(\sum_{x,y} log\Big(\delta + L_w(x,y)\Big)\right) \tag{6.6}$$

where $\bar{L}_w\Big(V(t)\Big)$ is the viewport key value at a given time t, $L_w(x,y)$ is the pixel luminance and N is the number of pixels in the viewport. δ is a small value to avoid singularity in the case the image contains black pixels. Here, time t does not correspond to a truthful timestamp, like in videos. In our case, t relates to an orientation of the camera due to the head movement of the user, which produces an image sequence in the viewport. To ensure a smooth transition between two successive viewports, the key value and the white value (i.e. the maximum luminance value in the viewport) are interpolated as:

$$\begin{aligned} \bar{L}_w^{'}(t) &= \tau \bar{L}_w\Big(V(t)\Big) + (1-\tau)\bar{L}_w^{'}(t-1) \\ L_{white}^{'}(t) &= \tau L_{white}\Big(V(t)\Big) + (1-\tau)L_{white}^{'}(t-1) \end{aligned} \tag{6.7}$$

where $\bar{L}'_w(t)$ and $L'_{white}(t)$ are respectively the smoothed key and white values between two successive views and τ is a time dependent interpolation variable. The value of τ determines the adaptation time. For $\tau = 1$ there is no adaptation, while for $\tau = 0$ the luminance is never updated. Based on TMOs that use models of eye adaptation [38], [39], we decided to fix the τ value corresponding to one second of adaptation (for both light and dark adaptations):

$$\tau = \Delta t \tag{6.8}$$

where Δt is the time spent between the previous and the current frame in second. Finally, the luminance is scaled and high values are attenuated to avoid clipping:

$$L(x, y, t) = \frac{a}{\bar{L}'_w(t)} L_w(x, y) \tag{6.9}$$

$$V(x, y, t) = \frac{L(x, y, t)\left(1 + \frac{L(x,y,t)}{L'^2_{white}(t)}\right)}{1 + L(x, y, t)} \tag{6.10}$$

where a is a user defined variable which scales the luminance (commonly 0.18) and $L(x, y, t)$ the time dependent scaled luminance.

Our Viewport TMO is the displayed luminance $V(x, y, t)$ as illustrated in Figure 6.5. In his operator, Yu uses Equation 6.9 that does not avoid clipping in high luminances. We have now the global coherency assured by the 360° image CDF ($G(x, y)$) and the viewport contrast ($V(x, y, t)$) we want to combine to obtain our final tone mapped image. The combination of both global and viewport TMO is explained below.

6.3.3 Tone Mapping Operators combination

The combination of global and viewport luminances ensures the global coherency to be preserved and the viewport contrast to be enhanced. To obtain a perceptually uniform resulting luminance, the combination has to be done in the logarithmic domain

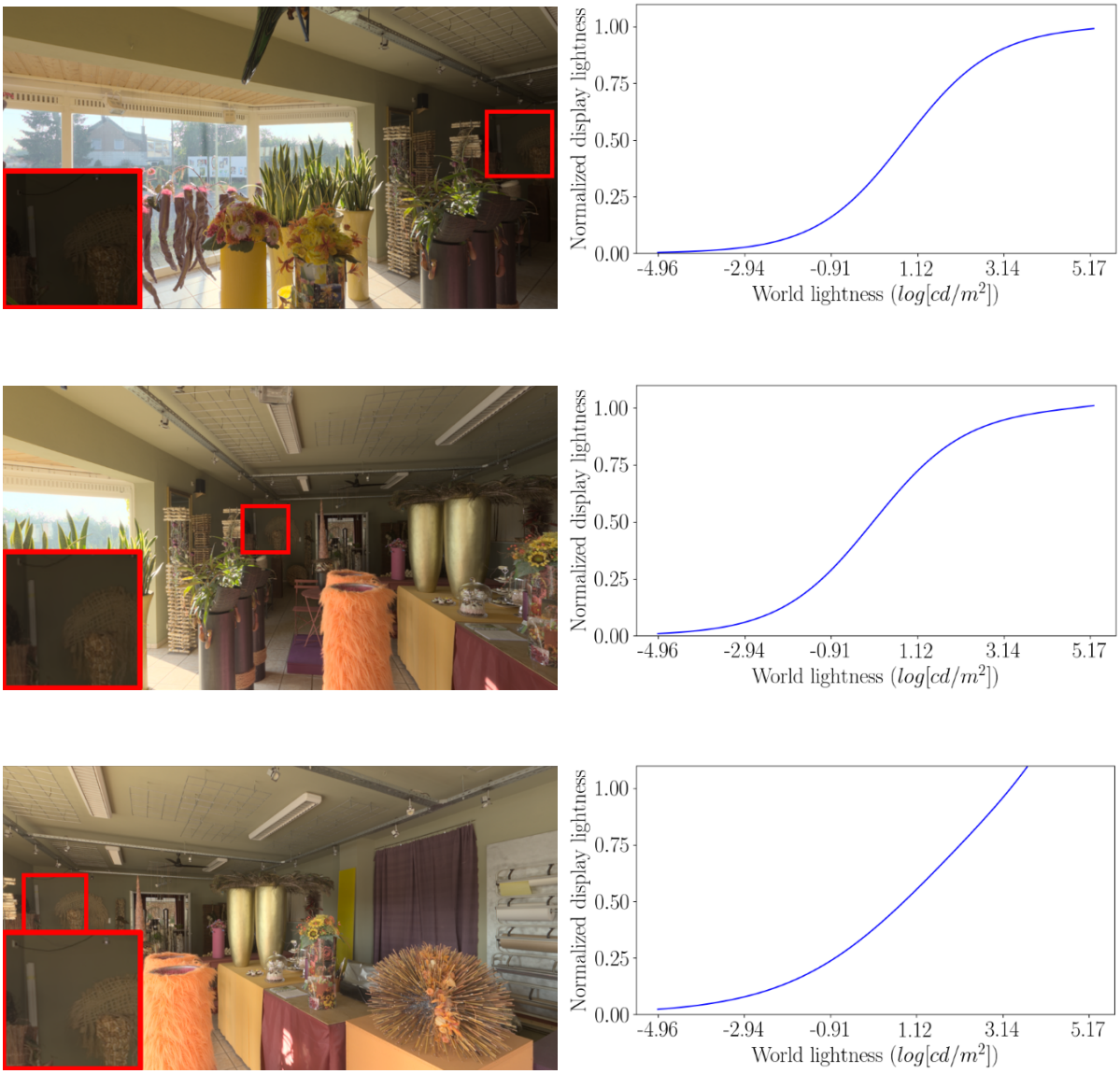


Figure 6.5: Photographic Tone Reproduction operator [85] applied to a viewport sequence with smooth transitions. As the key value and the white value evolve from a view to another, the tone curve is modified and the same zone in the scene (red inset) becomes brighter or darker.

as demonstrated in section 5.3:

$$\begin{aligned} ln\Big(L_d(x,y,t)\Big) &= \alpha.ln\Big(G(x,y)\Big) + (1-\alpha).ln\Big(V(x,y,t)\Big) \\ L_d(x,y,t) &= e^{\alpha.ln\Big(G(x,y)\Big)+(1-\alpha).ln\Big(V(x,y,t)\Big)} \\ L_d(x,y,t) &= e^{\alpha.ln\Big(G(x,y)\Big)} \times e^{(1-\alpha).ln\Big(V(x,y,t)\Big)} \\ L_d(x,y,t) &= \left(e^{ln\Big(G(x,y)\Big)}\right)^{\alpha} \times \left(e^{ln\Big(V(x,y,t)\Big)}\right)^{1-\alpha} \\ L_d(x,y,t) &= G(x,y)^{\alpha} \times V(x,y,t)^{1-\alpha} \end{aligned} \tag{6.11}$$

where $G(x,y)$ and $V(x,y,t)$ are our global and viewport TMOs respectively, α is a weight in the range $(0,1)$ that gives more emphasize on the global or the viewport result, and finally $L_d(x,y,t)$ is the display luminance. The effect of the α value is showed in Figure 6.6. As expected, the viewport TMO (left) enhances the contrast by exploiting all the dynamic range of the viewport. Contrarily, the global TMO (right) brightens a side of the 360° image and darkens another side, hence preserving the global coherency. The combination of both global and viewport TMOs (middle) preserves the global coherency of the scene while enhancing the viewport contrast. The value of α is not necessarily equal to 0.5 and can be adapted depending on the processed scene. More results are shown in Section 6.4.

6.3.4 Color saturation

Once our TMO has calculated the tone mapped luminance, we compute the color of all the pixels of the tone mapped image using the Schlick's approach [31]:

$$C' = \left(\frac{C}{L_w}\right)^{s} L_d \tag{6.12}$$

where C and C' are respectively the input and output trichromatic values (RGB), L_w the world luminance and L_d the tone mapped luminance. The saturation parameter s is set to 0.7 for our results. We used a basic method to manage the color, but several approaches of color correction exist for tone mapping [90], [91] and can be easily applied to this TMO. The efficiency of our TMO is illustrated in the following section.

Viewport TMO: $\alpha = 0$

Combination of both TMOs: $\alpha = 0.5$

Global TMO: $\alpha = 1$

Figure 6.6: While the viewport TMO ($\alpha = 0$) enhances the contrast in the image, the global TMO ($\alpha = 1$) preserves the global coherency of the 360° image. The combination of both TMOs ($\alpha = 0.5$) allows enhancing the image contrast while preserving the global coherency.

6.4 Results

We implemented our HMD-TMO using Unity3D [1] because of its friendly interface for managing VR and its capacity to handle HDR. We used the HTC Vive Pro [2] as an HMD. We benefited from GPU programming with shaders to compute 360° image histograms on the 2048×1024 equirectangular projection. The key values of both 1440×1600 viewports (left and right views for the binocular vision) are computed in real-time. Rendering (computation of the colored tone mapped image) is achieved with a shader applied to the HDR viewports. The global TMO is computed once for all in less than one second at the start. The navigation (calculation and display of successive viewport images) is performed in real-time: 90 frames are computed per second (Intel Core i7 vPro 7th Gen, NVidia Quadro M2200).

To evaluate the performance of our TMO against existing methods, we computed the Tone Mapped Image Quality Index (TMQI) [92] score on a dataset that consists of 90 views: 15 viewports from six different 360° HDR images. We calculated the average of the viewport scores for our method and three other TMOs. Overall, our method has the best mean TMQI score (see Table 6.1).

We also visually compare our results to those of Yu's [48] in Figure 6.7. In addition to preserving the global coherency, our TMO avoids clipping luminances out of the dynamic range of the HMD. This improvement is shown on the church wall in the first row images (green inset), and in the background and at the bottom left corner of the images in the second row. Furthermore, due to the exaggerated contrast produced by the CDF, our HMD-TMO enhances fine details. Indeed, in the first row, the contrast between the night sky and the tree leaves is higher with our method, which allows distinguishing holes through the foliage (red inset). The branches lying on the ground are also more detailed in the second row (green inset). The same phenomenon occurs in the images of the third row (red and green insets) and in the folds of the curtain in the images of the fourth row (red inset). However, an unwelcome effect appears in the last row. The lighting of the box

	Reinhard *et al.*	Ward *et al.*	Yu	Our
TMQI quality	0.798	0.854	0.865	**0.887**

Table 6.1: The result of the TMQI quality test: mean value computed on 90 images (Reinhard *et al.*'s [85] and Ward *et al.*'s [33] TMOs are applied to the entire 360° images).

1. `https://unity.com/`
2. `https://www.vive.com/fr/product/vive-pro/`

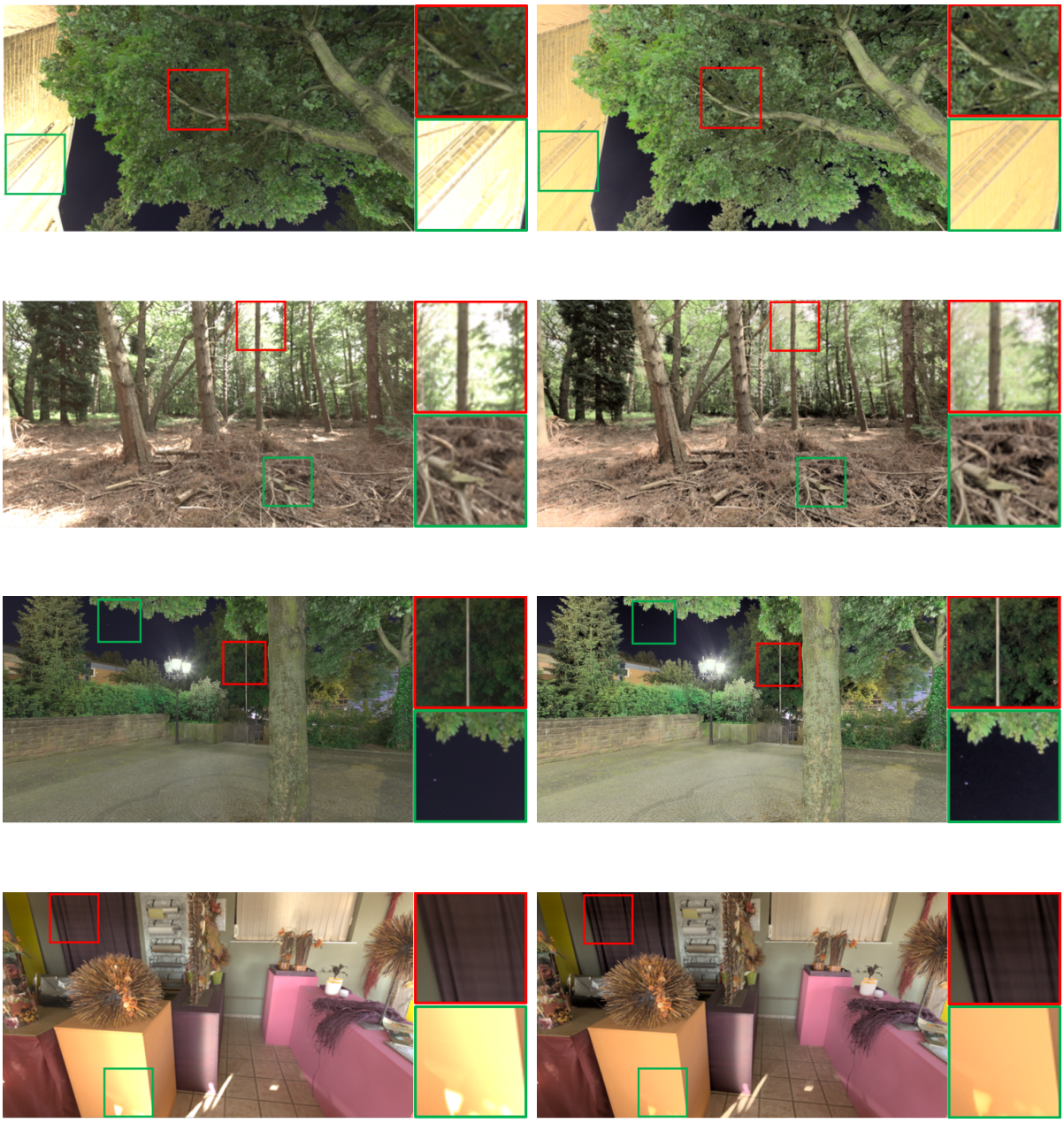


Figure 6.7: Yu's TMO [48] (left) compared to our HMD-TMO (right). Our method enhances fine details and removes the clipping in high luminance.

(green inset) almost disappears when using our TMO because its luminance is not enough represented by the CDF, which flattens this area.

Regarding the global coherency, the efficiency of our TMO is shown in the example presented in Figure 6.8. The two images on the top (6.8a) result from the viewport TMO. A little movement of the camera produces a significant change in displayed luminance, especially for the blue strip behind "Student Service". The bottom images (6.8c) result from the global TMO. The spatial coherency is preserved, the blue color stays the same. Nevertheless, many details are lost due to clipping in the top image. The two images in the middle (6.8b) result from the linear combination of both TMOs. The spatial coherency is well preserved, the blue strip does not change significantly and many details are visible in the top image. Figure 6.9 presents some additional results of our HMD-TMO.

6.5 Conclusion

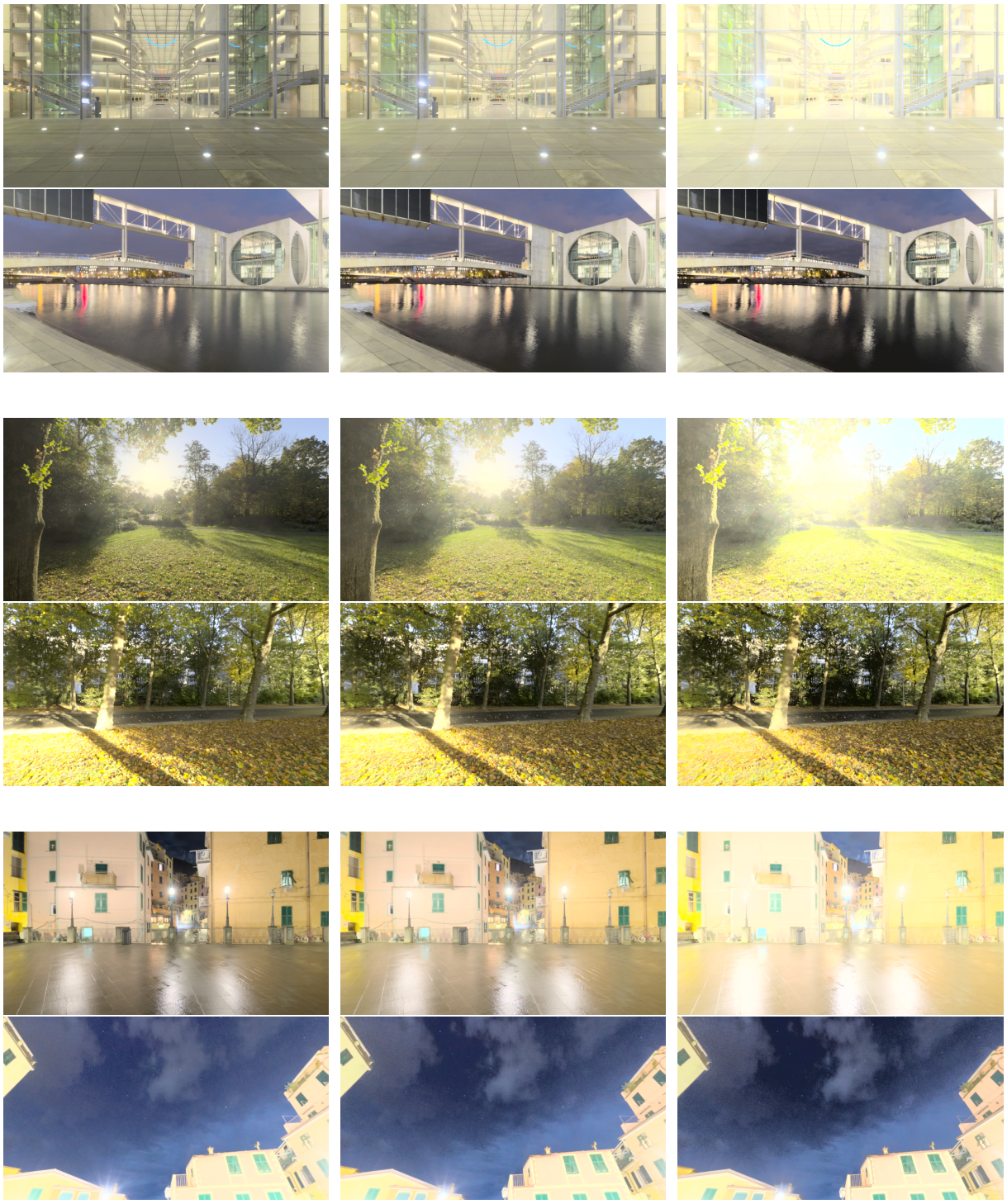
HDR imaging enables capturing the whole dynamic of a 360° scene. Previous subjective studies have shown that naive tone mapping of the entire 360° image or tone mapping of a viewport does not provide convincing results. To overcome these limitations, we have proposed HMD-TMO: a perceptually coherent TMO that combines both global and viewport TMOs. The linear combination in the logarithmic domain allows HMD-TMO to be adaptive depending on the encountered scene. This new TMO does not tackle the limits of tone mapping an image with a very high dynamic range but ensures a spatial coherency while navigating through the 360° HDR content. In the next chapter, we tackle tone mapping of interactive HDR 3D contents. The main challenges consist in accounting for temporal coherency, sudden changes in luminance range through time, and naturalness of time adaptation as detailed below.

(a) Viewport TMO: $\alpha = 0$

(b) Combination of both TMOs: $\alpha = 0.5$

(c) Global TMO: $\alpha = 1$

Figure 6.8: 6.8a Viewport TMO: The blue strip changes significantly. 6.8b TMOs combination: the spatial coherency and the details are preserved. 6.8c Global TMO: Some details are lost due to clipping.



(a) (b) (c)

Figure 6.9: (a) Viewport TMO: the local contrast is enhanced. (b) TMOs combination ($\alpha = 0.5$): the spatial coherency is preserved while the local contrast is enhanced. (c) Global TMO: the spatial coherency is preserved.

Chapter 7

A Tone Mapping Operator for High Dynamic Range 3D scenes with global lightness coherency

7.1 Introduction

In this chapter, we tackle real-time tone mapping of HDR rendering of 3D scenes taking into account both spatial and temporal coherency. Such applications, like video games, involve a camera moved by a user through a 3D scene. Each frame corresponds to the part of the scene facing the camera and rendered in HDR. The HDR image is then tone-mapped and displayed onto an SDR display (or an HMD). As described later, processing each image independently without accounting for spatial or temporal coherency may produce visual artifacts [43], which degrades the image quality. We propose a tone mapping method that accounts for the whole scene to ensure spatial and temporal coherency. This way, our TMO allows content consumers to better feel the lighting atmosphere of the scene and avoids visual artifacts such as flickering effects.

Furthermore, to demonstrate the efficiency of our TMO, we conducted a user study to evaluate the fidelity of tone-mapped content with respect to an HDR reference using different TMOs. Compared to other methods, we found that our TMO better preserves fidelity and is more appreciated. While our TMO works for interactive rendering of 3D scenes displayed on 2D screens, we focused on VR content because of the growth of this technology and the challenge of the existing tone mapping operators to target the limited dynamic range of today's HMDs. Moreover, in the VR use case, the user (and consequently the camera) is constantly moving, which increases artifact occurrence and degrades the immersion feeling. Finally, we propose a new subjective study protocol to compare an HDR content visualized on a 2D display with its tone-mapped version visualized on an HMD.

The chapter is organized as follows. Section 7.2 presents related work regarding real-time tone mapping. In section 7.3, we detail the changes we made to adapt our HMD-TMO to the problem of walking through 3D scenes rendered in HDR. The implementation of our TMO is detailed in section 7.4 while some results are presented in section 7.5. Finally, we describe the subjective study we conducted to evaluate our solution over previous TMOs in section 7.6. The chapter is then concluded in section 7.7.

7.2 Related work

The research in the field of temporally coherent TMOs started many years ago and has given rise to many different methods. Few TMOs simulate the visual adaptation over time to improve the feeling of contrast when visualizing a sequence of tone mapped images [38], [39]. Another way to improve the contrast for scenes with very high luminances is to simulate the maladaptation phenomenon [93]–[95]. This effect occurs when a visible area gets bleached after the user has looked at something very bright (the sun for example). Note that, in a bleached area, the color and the contrast are lost for few seconds.

More recently, many works have tackled TMOs for HDR videos [44]–[46]. Video TMOs have to cope with spatio-temporal coherency issues [43]. A complete survey on video tone mapping has been proposed by Eilertsen *et al.* [47], it especially details methods that take temporal coherency into account. For example, Zonal Brightness Coherency TMO proposed by Boitard *et al.* [44] consists in tone mapping a video by considering spatio-temporal segments to preserve the temporal coherency and avoid incoherent luminance changes. However, this solution cannot be used for real-time tone mapping of HDR images rendered from a 3D scene as the whole video sequence has to be known. Eilertsen *et al.* [46] proposed a real-time noise-aware tone mapping that answers the main issues of temporal consistency described by Boitard *et al.* [43]. Nevertheless, this TMO requires too much processing power to be processed in real-time and does not care about object brightness consistency.

On the other hand, our HMD-TMO described in the previous chapter addresses the issue of spatial coherency in the case of still HDR panoramas visualized on HMDs. In that case, the sequence of successive viewport images is unknown and the tone mapping has to be applied in real-time. By applying two different TMOs, a first one to the whole 360° image and a second to the viewport image, spatial coherency is preserved and the local contrast is enhanced. The viewport TMO is smoothed over time to mimic the eye

accommodation to the luminance level and consequently avoid the flickering effect produced by sudden changes in the image. The final tone mapped image is a combination of both TMOs in the logarithmic domain.

Durand and Dorsey [96] proposed an Interactive tone mapping that simulates the visual adaptation. This TMO is dedicated to interactive walk-through a 3D scene rendered in HDR and tone-mapped on the fly. They added a post-processing pass that accounts for chromatic adaptation and adds flares around light sources to increase the brightness effect. Due to visual adaptation, this TMO ensures temporal coherency but does not take into account the whole scene to tackle the spatial coherency.

Recall that our objective is to walk-through a 3D scene, render it in real-time from the current camera, tone map the resulting HDR viewport image (which corresponds to the current position of the camera), and display it on an SDR screen while preserving spatial and temporal coherency. Spatial coherency is obtained by accounting for the luminance of the whole scene. Temporal coherency is ensured by efficiently smoothing transitions between successive views. In this chapter, the image sequence is unknown because the camera is controlled by the user and the lighting conditions of the scene may change over time (turning on and off light sources, open doors, moving objects, etc.) Our solution relies on a combination of two TMOs as described in Chapter 6.

The first one is a global TMO that takes into account the luminance distribution of the whole scene to preserve the spatial and temporal coherency, while a second TMO, that only considers the image rendered on the viewport, enhances the local contrast and the temporal coherency. A more detailed description of our method is given in the following section.

7.3 Dynamic Range of a 3D scene

7.3.1 Problem statement

Our previous work about tone mapping 360° HDR images raises the issues of spatial and temporal coherency when these images are visualized on an HMD. At each time step, only a part viewport of the 360° image is displayed on the HMD, corresponding to where the user is looking at. To visualize these limited parts of the 360° image one after the other, a naive approach is to tone map each viewport image individually. This way, each current image in the viewport is well tone-mapped but the sequence visualized

as a whole can suffer from temporal incoherence. Indeed, the used TMO is based on a global operator proposed by Reinhard *et al.* [85] that computes luminance values of the HDR image: the log-luminance average and the maximum luminance value. If sudden changes in luminance happen from a view to another, the operation of tone mapping can produce flickering artifacts that break the temporal coherency. The solution initially proposed by Yu [48] is to smooth the tone mapping values over time to mimic the way the human eyes accommodate bright and dark luminances. In this way, flickering artifacts are avoided by the smooth transitions, and the temporal coherency is then preserved. Nevertheless, processing each image in the viewport irrespective of the entire 360° image tends to produce tone-mapped images with the same overall luminance average. Dark areas of the 360° image will be brightened while bright areas will be darkened. Thus, the spatial coherency of the 360° image is lost because all areas will be rendered with the same average luminance.

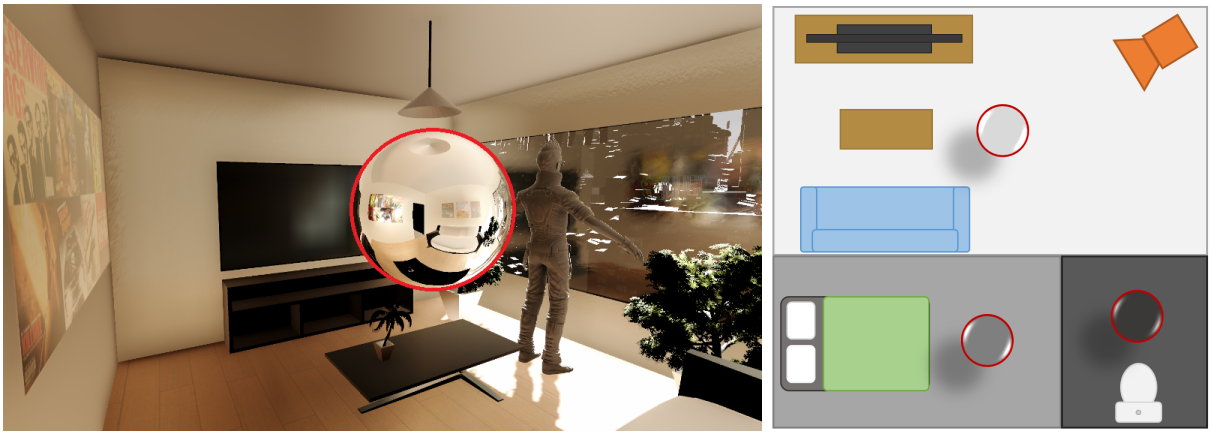
We proposed to solve the problem of spatial coherency by applying a TMO to the entire 360° image. Tone mapping the entire 360° image in one go with the same global operator preserves the spatial coherency. Dark areas are still dark on the tone-mapped image. Moreover, there is no more problem of temporal coherency (like flickering effect) as the tone mapping is applied once. Tone mapping the entire 360° image only once could solve the incoherence issues but it is to the detriment of reducing the used dynamic range of the tone mapped image in the viewport. Effectively, the dynamic range of a 360° image can be potentially much higher than a limited area of the same image in the viewport. Reducing the used dynamic range leads to a reduced contrast in the tone-mapped image that degrades the visual quality. We suggest keeping the advantages of those two TMOs by combining them.

Finally, a smoothed TMO applied to the viewport enhances the contrast and is temporally coherent while a TMO applied to the 360° image preserves the spatial coherency. Combining these two TMOs provides a balanced result. The global TMO can be processed once at the start and the viewport TMO is processed at each time step. To obtain a coherent tone mapped image, the combination is performed in the logarithmic domain (as explained in section 6.3). This approach can be reused in the case of interactive HDR rendering of 3D scenes. Indeed, the viewport TMO can be used as-is for each frame, while the global one has to be modified because the luminances of a 3D scene differs from the luminance of a 360° image. For this reason, HMD-TMO cannot be adapted to 3D scenes, it is then not evaluated in our study. The adaptation of HMD-TMO to 3D scenes is the

objective and the main contribution exposed in this chapter as detailed in the following section.

7.3.2 Approximating the luminance distribution

While the dynamic range of a still image has a clear definition (minimum and maximum luminance values), the dynamic range of an entire scene is a more complex notion. Looking at the luminance of all pixels of an image gives the dynamic range of this image. On the other hand, a straightforward approach to compute the dynamic range of a 3D scene is to look at the luminance of every space point of the scene. Indeed, for each 3D position (x, y, z) in the scene, if we compute the luminance coming from every direction (θ, ϕ), we can obtain the light field of the scene defined by the five-dimensional plenoptic function $L(x, y, z, \theta, \phi)$ [97]. Proceeding this way would give the most precise estimation of the luminance of the scene. In comparison, while the luminance of a pixel for an image is given by $L(x, y)$, the luminance at a position of a 3D scene that comes from a direction is given by $L(x, y, z, \theta, \phi)$. Nevertheless, computing such a luminance in a 3D scene leads to place a large number of 360° cameras at many points of the scene [98] and render the scene for all of these cameras, which is a complex, memory and time-consuming process. We assume that the luminance of the scene can be approximated by placing only a few 360° cameras at key positions as illustrated in Figure 7.1. The rendered images are 360° images



(a) (b)

Figure 7.1: (a) The environment map of the living room rendered by a 360° camera is represented by a sphere. (b) An example of a scene with three 360° cameras (circled in red) placed in the different rooms.

of the surrounding environment [99] called Environment Maps (EMs) from now on. Of course, the placement of the 360° cameras depends on the scene topology, and it is not excluded to put several cameras in the same room if this one is huge or with a particular topology. In the end, we get a set of HDR EMs rendered by all 360° cameras placed in the scene. An approximation of the dynamic range of the scene can then be given by looking at the luminance of all the pixels of those EMs. Nevertheless, the positioning of the 360° cameras may affect the luminance of the pixels, and then the result of the tone mapping, as detailed below.

7.3.3 Importance of the 360° cameras positioning

The positioning of the 360° cameras could affect the actual dynamic range of the scene. If the EMs are not placed correctly in the scene (for example not close to sources of high luminance such as light sources, specular and glossy surfaces, etc.) then the dynamic range of the scene is not well captured, which affects the TMO. We will precisely detail later how the TMO processes the rendered HDR image. In the example of Figure 7.2, when

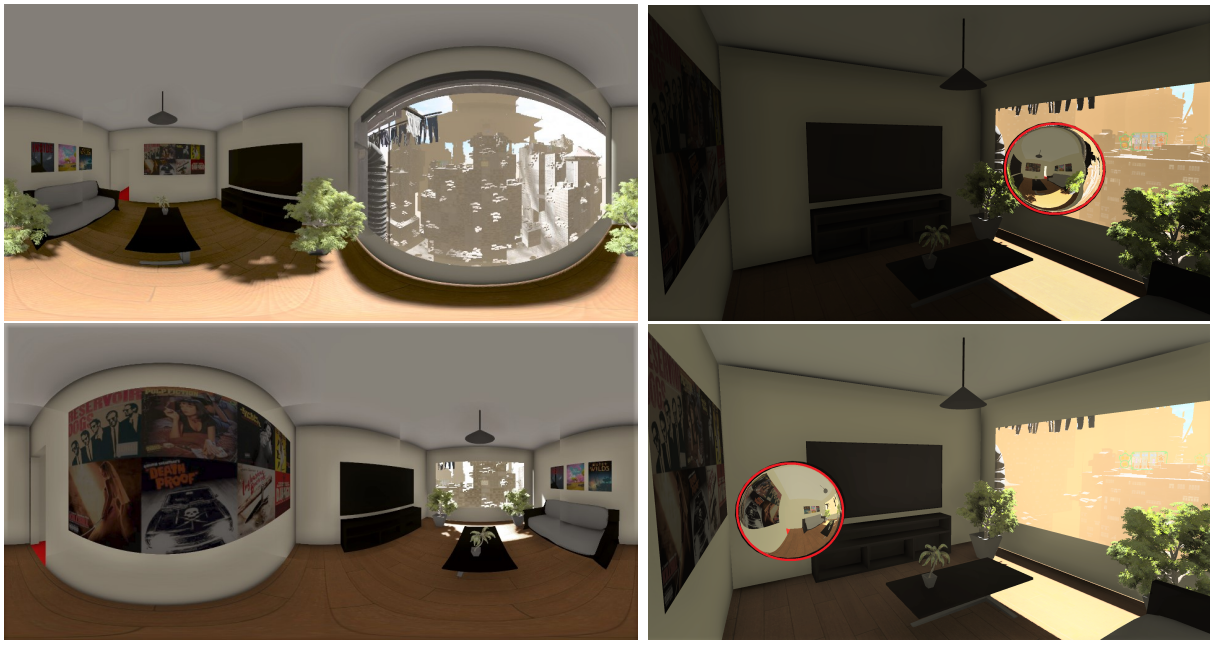


Figure 7.2: First row: the 360° camera is close to the window, the environment map (left) is mainly bright and the TMO darkens the image (right). Second row: the 360° camera is far from the window, the environment map (left) is mainly dark, the TMO brightens the image (right).

the 360° camera is placed close to a window (first row), the rendered EM is mainly bright (left image), consequently the tone mapped image is dark (right image). On the other hand, when the 360° camera is far from the window (second row), the EM is darker (left image) and the tone mapped image is bright (right image). In both tone-mapped images (right side), the position of the 360° camera is represented by a sphere circled in red. The placement of the 360° cameras in the scene has a real impact on the tone-mapped image and it is left to the content creator to place the 360° cameras depending on the expected result. Besides the influence of 360° cameras positioning, the rendering method used by these cameras to generate the HDR EMs can also impact the resulting tone mapping.

7.3.4 Comparison of rendering methods

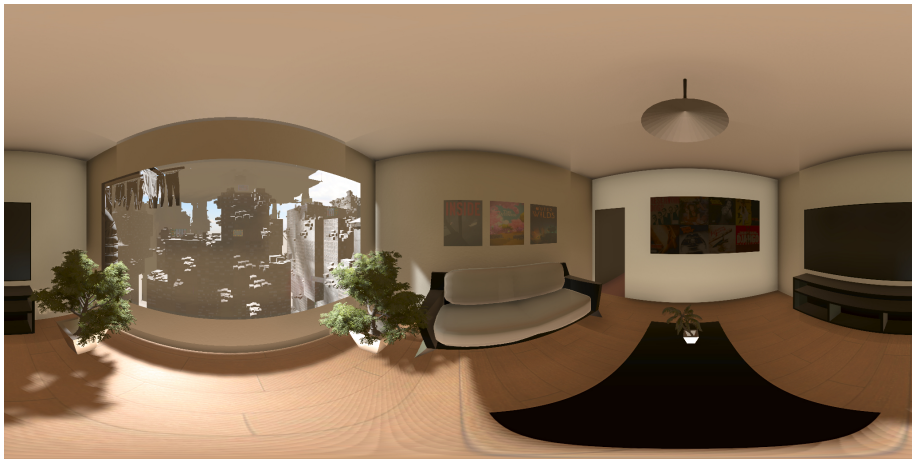
To obtain the most representative luminance distribution of the scene, a realistic simulation of the light interaction with objects surfaces should be done. Ray tracing-based methods produce more realistic images but are still too slow to render multiple EMs of complex 3D scenes in real-time. Nevertheless, approaching a realistic rendering in real-time can be achieved using some simplifications. We decided to render direct lighting and add ambient light and ambient occlusion terms that coarsely approach indirect lighting.

The rendering performed by these two methods (ray tracing and ambient-based rendering) produces slightly different results that are illustrated in Figure 7.3. Moreover, as detailed later, only the log-luminance average value ($\bar{L}_w$ in equation 6.6) of the EMs is needed to compute our global TMO. We compare the log-luminance average of the environment map rendered using ray tracing (Figure 7.3a) and when approximating it with ambient and occlusion terms (Figure 7.3b). The difference between these two computed values ($\bar{L}_w$) does not severally impact the resulting TMO. Thus, all EMs are rendered in real-time by computing direct lighting added to the ambient and occlusion terms. Our global TMO is then performed using all the rendered EMs as described in section 7.4. Finally, as the process has to be real-time, we propose three optimizations that can increase significantly the processing time of rendering.

7.3.5 Optimizations

First, we propose to render the EMs with a low resolution as only the log-luminance average value is needed by the global TMO. This value is pretty equivalent for a high or a reasonably low resolution.

(a) Ray tracing rendering: $\bar{L}_w = 0.012$.



(b) Ambient-based rendering: $\bar{L}_w = 0.01$.

Figure 7.3: The log-luminance average between (a) ray-tracing rendering and (b) ambient-based rendering does not differ enough to significantly impact the result of the TMO.

Secondly, the number of EMs to render impacts the processing time. As the scene can be huge, dozen of 360° cameras can be useful to obtain the global TMO. We propose to limit the number of EMs to render by considering only the nearest 360° cameras from the user. The number of 360° cameras to consider is a user-defined parameter. The impact of the number of EMs to render and their resolutions is shown in Figure 7.4.

Moreover, the scene is interactive: lights can be turned on and off, doors can be opened and closed, etc. To obtain a correct global TMO, EMs should be rendered at each frame, which is time-consuming. We finally propose to render the EMs only at the first frame, and when the scene changes. A change is detected only for objects that can significantly affect the luminance of the scene (i.e. lights and doors, moving little objects does not

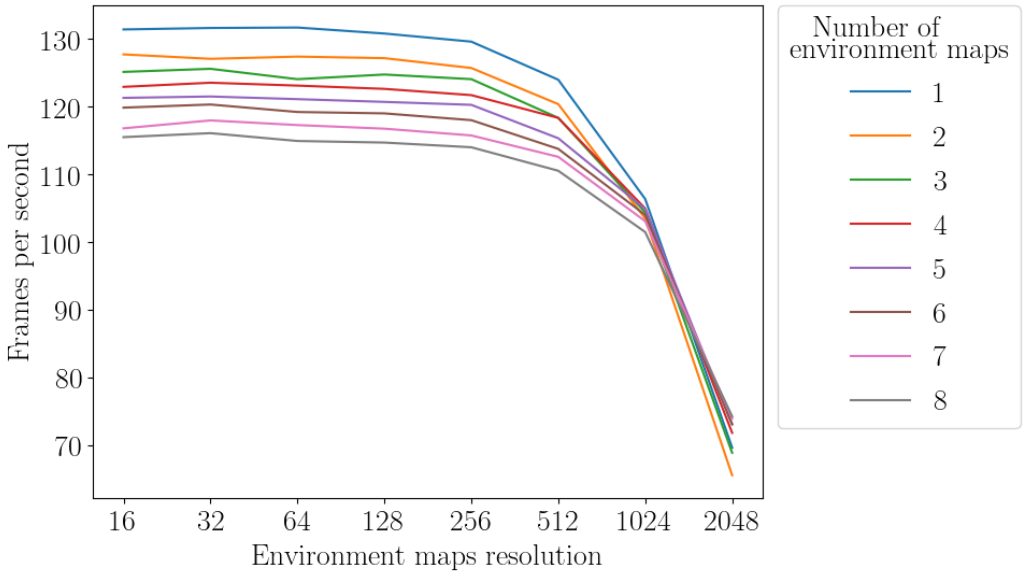


Figure 7.4: Impact of the number of EMs to render and their resolution for a complex 3D scene (CPU Intel core i5 6600 3.30GHz, 16GO RAM, NVidia GTX 980Ti).

impact the luminance too much). If any change is detected on an object close to a 360° camera, the EM is rendered again to take into account this change in real-time.

7.4 Tone Mapping Operator for High Dynamic Range rendering of 3D scenes

7.4.1 Global Tone Mapping Operator of a 3D scene

As a reminder, the global TMO performed by HMD-TMO relies on the log-luminance histogram of the 360° image. The computation of the histogram and the ceiling process detailed in Section 6.3 is time-consuming and is performed once at the start. In our current case, the global TMO cannot rely on the histograms of the EMs because it has to be computed at each frame. To do so, for our global TMO, we decided to use the viewport TMO applied in real-time in HMD-TMO. To recall, this TMO relies on the Photographic Tone Reproduction operator proposed by Reinhard *et al.* [85] and is defined in Section 6.3. However, in our current case, the log-luminance average $\bar{L}_w$ (defined in equation 6.6) can not be computed in the same way as the scene is not represented by a unique 2D image but by a set of HDR EMs. The computation of our global TMO is detailed below.

First, we compute the log-luminance average $\bar{L}_w$ of each HDR EM_i rendered by the 360° cameras:

$$\bar{L}_w\Big(EM_i\Big) = \frac{1}{N_i} exp\left(\sum_{x,y} log\Big(\delta + L_w(EM_i, x_t, y_t)\Big)\right) \tag{7.1}$$

where L_w is the world luminance of a pixel (x_t, y_t) of the EM_i at a time t. A small value δ is still used to avoid singularities and N_i is the number of pixels for the EM_i. The global log-luminance average of the whole scene is then approximated by the probability for the user camera to be in a particular room at a moment. As we do not have any a priori on the position of the user in the scene, the probability of being in a room is the same everywhere. This leads to an arithmetic mean of the log-luminance averages of all EMs:

$$\bar{L}_w = \frac{1}{M}\left(\sum_i \bar{L}_w\Big(EM_i\Big)\right) \tag{7.2}$$

where M denotes the number of 360° cameras that are considered to compute the global TMO. Then, the global TMO processes each pixel (x_d, y_d) of the viewport as follows:

$$L(x_d, y_d) = \frac{a}{\bar{L}_w} L_w(x_d, y_d) \tag{7.3}$$

$$G(x_d, y_d) = \frac{L(x_d, y_d)}{1 + L(x_d, y_d)} \tag{7.4}$$

where a is a user-defined value (default value is equal to 0.18) and G is the global display luminance.

At this point, the spatial coherency is preserved as the TMO is computed considering the log-luminance average of the entire scene. However, temporal changes are not considered because luminance variations in the scene can happen (lights turn on and off, doors open and close, etc). To take into account these variations, the computation of the log-luminance average of the whole scene (equation 7.4) is performed at each time step. Indeed, if an area of the scene suddenly becomes very bright, the scene luminance globally increases, and the tone mapped image is darkened. To avoid a sudden change in the tone mapped image, we decided to temporally smooth the value of the log-luminance average of the whole scene in the same way we smoothed the viewport TMO in HMD-TMO:

$$\bar{L}'_w(t) = \tau_g \bar{L}_w(t) + (1 - \tau_g)\bar{L}'_w(t-1) \tag{7.5}$$

where $\bar{L}_w^{'}(t)$ is the smoothed log-luminance average value of the entire scene between two successive time steps and τ_g is a time-dependent interpolation variable which defines how fast is the smoothing. We fixed $\tau_g = 0.1$ in our implementation, which means the transition takes about 10 seconds. The smoothed log-luminance average value $\bar{L}_w^{'}(t)$ replaces $\bar{L}_w$ in equation 7.3. This allows us to adapt the tone mapping to temporal changes of lighting in the scene with smooth transitions.

7.4.2 Viewport Tone Mapping Operator of the user camera

As detailed in HMD-TMO, a TMO only applied to the global scene results in very limited use of the dynamic range of the display. To use a higher dynamic range and then enhance the contrast in the tone-mapped image, a second TMO has to be applied to the viewport. We propose to compute a second tone mapping of the image rendered in the viewport and combine it with the global TMO. The used operation is the same, but the log-luminance average is that of the HDR image of the viewport.

$$\bar{L}_w = \frac{1}{N} exp\left(\sum_{x_d, y_d} log\big(\delta + L_w(x_d, y_d)\big) \right) \tag{7.6}$$

$$\bar{L}_w^{'}(t) = \tau_v \bar{L}_w(t) + (1 - \tau_v)\bar{L}_w^{'}(t-1) \tag{7.7}$$

$$L(x_d, y_d) = \frac{a}{\bar{L}_w^{'}} L_w(x_d, y_d) \tag{7.8}$$

$$V(x_d, y_d) = \frac{L(x_d, y_d)}{1 + L(x_d, y_d)} \tag{7.9}$$

where L_w is the world luminance of the pixel (x_d, y_d) in the viewport image, $\bar{L}_w$ is the viewport log-luminance average, and a is still the user defined value. $\bar{L}_w^{'}(t)$ is the smoothed value between two successive views and τ_v is a time dependent interpolation variable ($\tau_v = \Delta t$ to have an adaptation of about 1 second as defined in HMD-TMO). Finally, V is the viewport tone-mapped luminance. There, the temporal coherency is also preserved by smoothing the $\bar{L}_w$ value over time (equation 7.7). Besides, the eye temporal adaptation is implicitly involved in these smooth transitions.

7.4.3 Resulting Tone Mapping Operator: Combination of global and viewport Tone Mapping Operators

Finally, the combination of both, global and viewport TMOs is performed in the logarithmic domain as defined in HMD-TMO and reminded below:

$$L_d = exp\Big(\big(\alpha \times ln(G)\big) + \big((1-\alpha) \times ln(V)\big)\Big) \tag{7.10}$$

$$L_d = G^{\alpha} \times V^{1-\alpha} \tag{7.11}$$

where G and V are respectively the resulting luminance of the global and the viewport TMO. α is a variable in the range $(0, 1)$ that gives more emphasis on the global or the viewport result and L_d is the final display luminance. Obviously, L_d, G and V are evaluated for each pixel (x_d, y_d).

7.4.4 Colorization

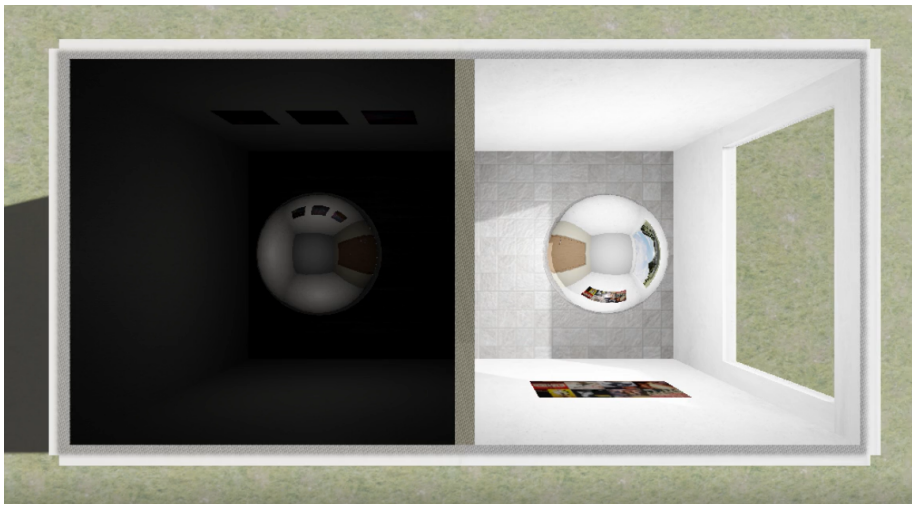
During the entire process, only the luminance is managed. The color is added at the end using the Schlick's approach [31]:

$$C' = \left(\frac{C}{L_w}\right)^s L_d \tag{7.12}$$

where C and C' are respectively the input and output trichromatic values (RGB), L_w the world luminance and L_d the tone mapped luminance. The saturation parameter s is set to 0.7 for all of our results. The efficiency of our TMO is illustrated in the following section.

7.5 Results

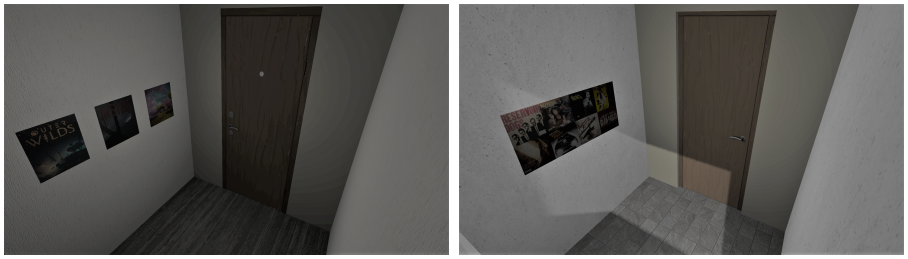
In this section, we evaluate our TMO for scenes rendered in HDR. We compare tone-mapped images either rendered from two different viewpoints of the scene or from a different time with a camera that moves through the scene. We start by giving a detailed example of the effect of combining both TMOs illustrated by Figure 7.5. In this example, we compare two rooms of the same scene (top image). One is very dark (left column)



Top view of the scene showing the two rooms: the dark one on the left and the bright one on the right.

$\alpha = 1$ (Global TMO)

$\alpha = 0.5$



$\alpha = 0$ (Viewport TMO)

Figure 7.5: The α variable of the combination (equation 7.11) gives more emphasize on the viewport TMO or on the global TMO. The left column and right column are respectively the tone-mapped renderings of the dark room and the bright room in the same scene showed by the top image.

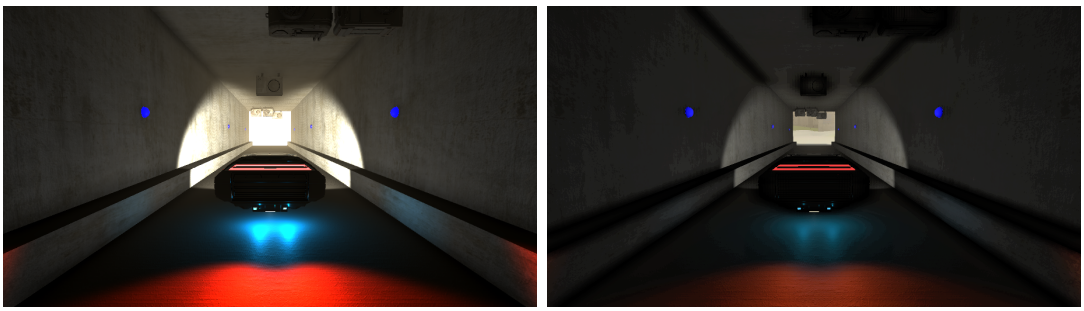
and the second is very bright (right column). When considering the global TMO only ($\alpha = 1.0$), the global contrast between the two rooms is well preserved. However, details in both images are lost. The light reflection on the wall in the bright room is clipped, the floor in the dark room is totally black. Regarding the result when considering the viewport TMO only ($\alpha = 0.0$), the contrast in each image is enhanced, all details are visible. However, the global coherency is completely lost, both rooms seem equally lit, it is impossible to say if one room is brighter than the other. A geometric mean between the two TMOs ($\alpha = 0.5$) gives a good compromise for both images. The images of both bright and dark rooms are well detailed and the global contrast is preserved.

We also compared our method with a TMO applied to the viewport only with smooth transitions between views that simulate the visual adaptation to the light [1]. We are interested in temporal coherency as illustrated in Figure 7.6. This example corresponds to a scene where a car is driving in a tunnel. With the naive TMO only applied to the viewport (left column), the temporal coherency is lost. At the start ($t = 0$), the car is inside the tunnel, only the spotlights are illuminating the area. We see in the background that the end of the tunnel is completely clipped. At the end ($t = 3$), the car is outside the tunnel, the image is mainly bright. We can not see anymore the red and blue lights from the back of the car. Over time, the HDR image became brighter and brighter, which leads to a darker and darker tone mapped image. With our TMO (right column), the inside and the outside of the tunnel are visible at each time in the tone-mapped image because we take into account the luminance of the whole scene. Even if the contrast is reduced in each image independently, the temporal coherency is globally preserved.

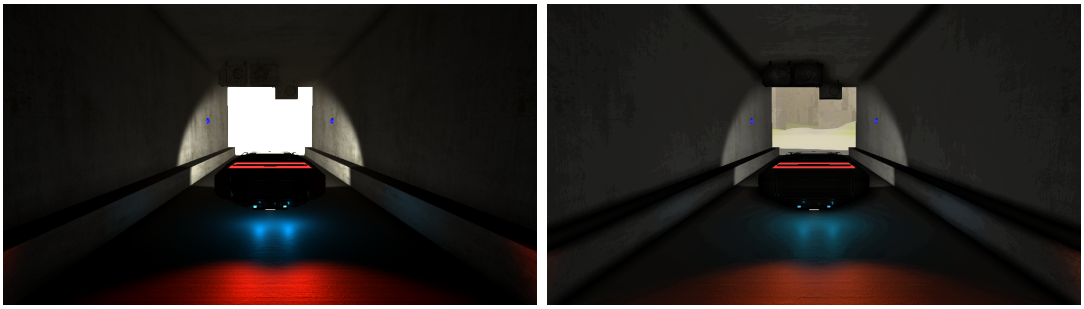
7.6 Subjective evaluation

In this section, we detail our evaluation protocol regarding the efficiency of our TMO compared with two others. We suppose that our method is more faithful with respect to the HDR content and that users appreciate more our TMO. To confirm that, we conducted a subjective evaluation where participants had to judge an HDR content tone mapped with three different TMOs compared with the same HDR content visualized on an HDR display (the ground truth). Our panel consists of 18 participants (9 females and 9 males), between 23 and 44 years old. 9 were experts in VR with daily use while the others had

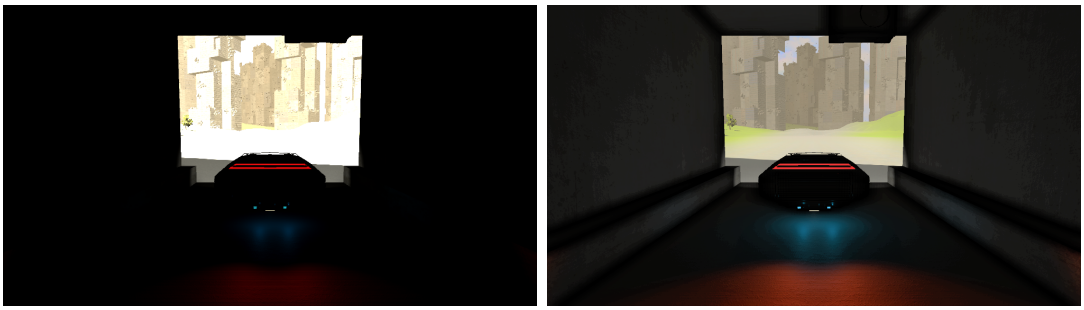
1. Eye-adaptation TMO implemented in Unity3D (`https://docs.unity3d.com/560/Documentation/Manual/PostProcessing-EyeAdaptation.html`)



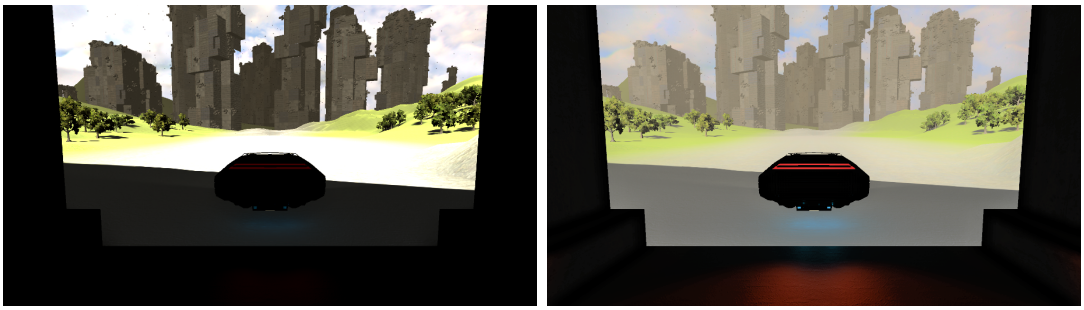
$t = 0$



$t = 1$



$t = 2$



$t = 3$

Figure 7.6: Comparison between eye-adaptation (left column) and our TMO (right column) on an image sequence. The eye-adaptation TMO only applied to the viewport does not preserve the temporal coherency. The tunnel is less and less visible while the outside becomes visible when the car gets out. Our method preserves the temporal coherency, both inside and outside of the tunnel are always visible.

only a few VR experiences before the experiment.

7.6.1 Experiment protocol

The objective is to allow a user to compare scenes visualized on an HDR display (the ground truth) with the tone mapped result visualized on an HMD.

Evaluating TMOs

As explained before, HMD-TMO cannot be evaluated in our study. Even if HMD-TMO ensures visual coherency while displaying 360° HDR images, this coherency is limited to head rotation movement while in our case we tackle walk-through 3D scenes. That is why we choose to evaluate our TMO by comparing it with two others that can be computed in real-time. The first one is a naive method, denoted Logarithm, that is based on the Adaptive logarithmic mapping for displaying high contrast scenes proposed by Drago *et al.* [34] with smoothed values over time to avoid flickering artifacts. In the implementation of this first TMO, the b factor defined in the related paper is set to 0.8 as suggested by the authors. The second TMO is the eye-adaptation TMO implemented in Unity3D. This TMO is based on the Photographic tone reproduction for digital images [85] with smoothed values too. In addition to ensuring temporal coherency, this implementation tries to preserve the global coherency by introducing a scale factor. This parameter applies an offset to the display luminance depending on the world log-luminance average of the current image. In other words, if the HDR image is globally dark, this method applies a negative offset to darken the displayed image. It works similarly for bright images. This method is effective for a user-defined luminance range. For our purpose, we fixed the low and high limits corresponding to the minimum and maximum luminance values that can be encoded by the used HDR display (ie. the dynamic range of the SIM2 display goes from 10^{-4} to $10^{4} cd/m^{2}$). To sum up, the three TMOs used in the experiment are:

1. Our TMO described in section 7.4
2. Eye-adaptation: The default eye-adaptation TMO implemented in Unity3D
3. Logarithm: Adaptive logarithmic mapping [34] with smoothed values to avoid flickering.

Evaluation criteria

We asked participants to evaluate the three TMOs regarding two aspects. First, they had to grade the visual fidelity of the tone-mapped content visualized on the HMD compared with the HDR content visualized on the HDR screen. Then, we ask them to rank their global appreciation regarding the visual quality of the tone-mapped content regardless of the HDR reference. Finally, they had to rank the three TMOs by order of preference.

Evaluating 3D scenes

To evaluate our TMO, we developed two interactive scenes in Unity3D. These scenes are interactive as the camera can move and a door can be opened or closed. Both scenes have very contrasting areas (very dark and very bright) to confront evaluated TMOs with difficult conditions. The first one takes place in a car and the scene consists in driving of a tunnel. Outside the tunnel, the sun illuminates the scene while inside the tunnel, the car's headlights are the only light sources as shown in Figure 7.6. The second scene takes place in a little apartment. One room has a huge window and is illuminated by the sun. The second room is totally closed and is illuminated by weak light. The two rooms are separated by a door as illustrated in Figure 7.5.

Evaluation procedure

The procedure can be described as follows. We start by explaining to participants the test protocol and ask them to fill in the consent form. First, we ask them to interact with the HDR scene for few minutes in order to view the different areas of the scene. The interaction with the scene is very simple: participants use a mouse to manipulate the position and the orientation of the camera in the scene. The content visualized on the HDR display is the ground truth.

Once the HDR scene has been explored, we present to participants the same scene on the HMD. In that case, participants move physically in rotation and/or translation depending on the tested scene. The scene of the apartment had the same dimensions that the physical experiment room, so participants are entirely free of movement. For the scene of the tunnel in the car, participants are sat at the driver position and we give them a mouse to move the car forward and backward (by pressing left and right buttons). A constant speed was applied to the car in order to reduce the effect of motion sickness

experienced by users [100], [101]. We observed this effect thanks to a Simulator Sickness Questionnaire (SSQ) [102]. The image on the HMD is tone mapped using one of the three evaluated TMOs.

Finally, when the HDR content and the tone mapped content have been seen, we ask participants to give their grades regarding the two criteria: (1) The fidelity of the tone-mapped content compared with the HDR reference is graded using a linear scale between 0 (not faithful at all) and 10 (very faithful). (2) The global appreciation regarding the visual quality of the tone-mapped content is graded using a linear scale between 0 (not appreciated at all) and 10 (very appreciated). To sum up, a protocol loop consists of three steps:

1. Interact with the HDR content on the SIM2 HDR screen
2. Interact with the tone mapped content on the HMD
3. Grade the visual fidelity and the global appreciation

Once the three TMOs for a scene have been tried, we also ask participants to rank the three TMOs from the best to the worst in order of preference. For the whole study, we evaluate three TMOs for two scenes, which means the participants follow the protocol six times in total. We also randomized the order scenes and TMOs are tested to avoid any bias.

Software and hardware material

The hardware used was a SIM2 HDR display for the ground truth and an HTC Vive pro as an HMD. To render HDR images, an image effect shader compliant with the SIM2 display has been implemented. We implemented our solution in Unity3D as it is a friendly tool for VR and HDR development. Tone mapping values are computed in real-time with compute shaders. Then we apply our TMO as an image effect shader on the rendered HDR images. All the process has to be real-time and run at least at 90 frames per second to ensure a pleasant experience in case of VR. We will now present the results of the experiment.

7.6.2 Experiment results

As our experiment involved physical navigation as well as virtual navigation using a controller, users fulfilled the Simulator Sickness Questionnaire (SSQ) before and after the test. Indeed, cybersickness could have an impact on our results. We followed the approach

proposed by Kennedy [102] to compute the SSQ score of each user before and after the experiment. The mean (M) and standard deviation (SD) scores of the SSQ were slightly greater at the end of the experiment (start: $M = 11.01$, $SD = 15.04$; end: $M = 14.13$, $SD = 21.75$) mainly due to the disorientation symptoms. However, these scores can be considered relatively low. As these scores do not follow a normal distribution, we used a Wilcoxon Matched Pair Signed Rank Test to determine if this difference was significant. The result suggests that our experiment did not have a significant impact on cybersickness ($p > 0.05$).

Regarding the different scores given for fidelity and appreciation, we first analyzed both criteria for each scene (scores presented in Table 7.1 and 7.2). Globally, our TMO obtained the best scores while Logarithm obtained the worst for both fidelity and appreciation criteria. However, we performed an analysis of variance (ANOVA) on these results but we did not find any significant difference.

Results were also analyzed independently from the scene, by adding the scores obtained for both scenes for each criterion: fidelity and appreciation. We also performed an ANOVA on these results and we found a significant difference for the fidelity criteria ($p < 0.05$) as represented in Figure 7.7. Then, for comparing each pair of conditions, we performed a post hoc Turkey test and we only found a significant difference between Logarithm and our TMO conditions ($p < 0.05$). No significant difference was found for global appreciation.

Regarding the ranking of the three TMOs given at the end of the experiment, results are shown in Figure 7.8. Whatever the scene, our TMO (green) is the one that is the most ranked first and second and the least ranked third. The Logarithm TMO (grey) is

Fidelity

	Logarithm	Eye-adaptation	Our TMO
Scene 1	6.39 ±1.82	6.56 ±1.82	**7.50** ±1.15
Scene 2	6.17 ±1.69	7.00 ±1.64	**7.39** ±1.58

Table 7.1: Means and standard deviations of the fidelity for the different TMOs.

Global appreciation

	Logarithm	Eye-adaptation	Our TMO
Scene 1	6.83 ±2.20	6.94 ±1.66	**7.44** ±1.38
Scene 2	6.28 ±1.81	7.22 ±1.86	**7.61** ±1.65

Table 7.2: Means and standard deviations of the global appreciation for the different TMOs.

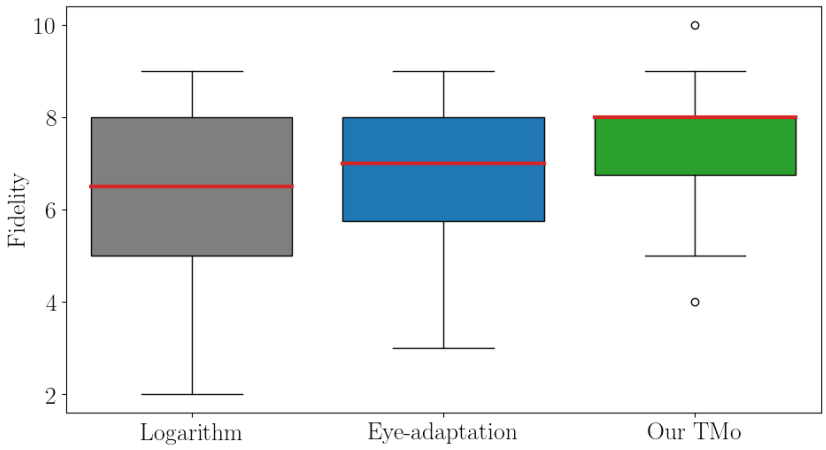


Figure 7.7: Box plot for the fidelity scores overall.

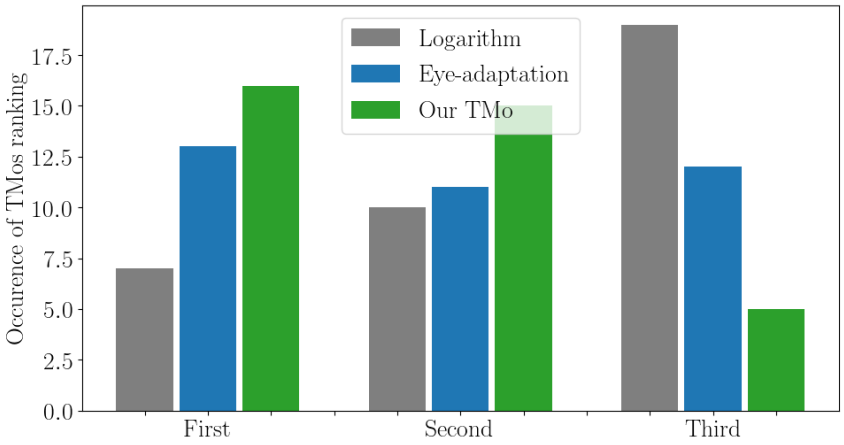


Figure 7.8: Occurrence of TMOs ranked first, second and third overall.

often ranked third while the Eye-adaptation TMO (blue) is equally ranked first, second and third. Indeed, our TMO tends to be the one that satisfies most users.

To sum up, the non-significant results do not allow us to give a clear conclusion. However, they suggest that our method is globally more faithful and more appreciated than the others, especially compared with the Logarithm method.

7.7 Conclusion

Rendering in computer graphics produces more and more HDR images. Preserving the artistic intent and lighting atmosphere of the original content is essential during the tone mapping process. Previous methods do not consider the whole scene to tone map

rendered images, which causes a loss of global coherency. To overcome this limitation, we proposed to consider the whole scene to compute a global tone mapping combined with a viewport tone mapping to render our final image. The entire process is performed in real-time and can be applied to more complex scenes. We also showed that our method is slightly preferred when compared with other ones.

Part IV

Point cloud stylization

Introduction

In this part, we propose a new pipeline with methods adapted from the image domain to automatically transfer the color from a target point cloud to an input point cloud. These color transfer methods are based on color distributions and account for the geometry of the point clouds to produce a coherent result. The proposed methods, relying on simple statistical analysis, are effective and succeed in transferring the color style from one point cloud to another. The qualitative results of the color transfers are evaluated and compared with existing methods.

Chapter 8

Example-based color transfer for 3D point clouds

8.1 Introduction

In this chapter, we focus on color transfer between 3D point clouds. To the best of our knowledge, the only existing attempt to perform a color transfer between 3D point clouds [103] is a CNN-based method originally used for point clouds classification [104]. We propose new color transfer methods that account for both the color and the geometry of the 3D point clouds. Our methods rely on the matching of color distribution variances and assume that color channels follow Gaussian distributions. For our purpose, accounting for the geometry of the 3D point clouds is essential, especially the normals to the surfaces. Before performing the color transfer, the normals of the input and the target point clouds are projected into the best fitting 3D coordinate system by a Principal Component Analysis (PCA) to improve the quality of the transfer.

The chapter is organized as follows. Section 8.2 reviews previous work on image style transfer and point cloud rendering. Then, we present our color transfer pipeline for point clouds (considering both the color distributions and the point clouds geometry) and detail our two distribution-based methods in Section 8.3. The results of our methods and different applications are presented in Section 8.4. Thereafter, we evaluate the performance of our methods and compare them with other techniques that consider or not the geometry. We finally conclude the chapter in Section 8.5.

8.2 Related work

Color style transfer for 2D images. Over the past years, image style transfer aroused great interest. Reinhard *et al.* [105] proposed a color transfer between images based on a simple statistical analysis. Color signals are supposed to follow Gaussian distributions

and to be independent of each other as they are first converted into the decorrelated $l\alpha\beta$ color space [106]. The transfer is then achieved by matching the distributions of the target colors with the distributions of the input colors, as detailed in Section 8.3.3.

Later, Pitié and Kokaram [107] relaxed the independence assumption and supposed that the color signals follow a Multivariate Gaussian Distribution (MGD). The transfer is achieved using a linear transformation defined by the Monge-Kantorovich closed-form matrix. The authors evaluated several color spaces and obtained better results when using the CIELAB space. This transformation is also detailed in Section 8.3.3.

Hristova *et al.* [108]–[110] went further by considering the color channels as a Multivariate Generalized Gaussian Distribution (MGGD) or as a beta distribution. Relaxing the shape parameter of the MGGD allowed them to improve the style transfer between images and even proposed to perform an n-dimensional style transfer. A 5-dimensional transfer of image colors and gradients is given as an example in their work [109].

More recently, Convolutional Neural Networks (CNNs) have proved their efficiency in addressing style transfer problems. Gatys *et al.* [111] used a CNN optimized for object recognition to separate the semantic content of images from the style. In their method, high-level features, corresponding to the content, are extracted from the input image while texture information of the target image is given by a Gram matrix. An iterative process of gradient descent gives the final stylized image that results from a linear combination of the content and the style features. Johnson *et al.* [112] achieved similar qualitative results with a faster method. Furthermore, CNN-based style transfer has been extended to different domains, like fluids simulations [113] or interactive videos [114].

Style-based rendering of 3D point clouds. While many research works seek to improve the quality of point clouds rendering (such as holes filling [69]–[71], reducing edges aliasing [115], drawing curved shapes instead of circles as a primitive [67]), only a few attempts exist about point clouds stylization.

Xu *et al.* [116] proposed to render point clouds silhouette to get a cartoon style. Their method is based on two rendering passes. Points are firstly rendered in black color with a large radius while the second pass performs a rendering of the points with their appropriate colors and radius. Flat surfaces are then filled in with the right content while the edges are surrounded by a black silhouette.

A few years later, Rosenthal and Linsen [6] improved point clouds rendering in the image space. A point cloud is rendered before proceeding to holes filling, edge detection,

and anti-aliasing in the image space. Silhouette rendering is then a direct application of edge detection (see Figure 1.6).

Color style transfer for 3D point clouds. After a first attempt in transferring textures style for 3D models [117], a first study finally focused on point clouds style transfer. Cao *et al.* [103] proposed $PSNet$, a color transfer network for point cloud stylization. Based on a network trained for point clouds classification and segmentation [104], they capture high-level features [118] to transfer either the color or the geometry (or both) of a target point cloud to an input point cloud. Regarding color transfer, they also add the possibility to consider a simple image as a target where each pixel corresponds to a 3D point. As in [111], Cao *et al.* used the Gram matrix of features map to represent the color style of the target.

Nevertheless, $PSNet$ is efficient for point clouds similar to those used by its training dataset [119], which consists of single objects across 16 categories (lamp, chair, table, etc.), while our methods can be used regardless of the nature of the point clouds. We show later, by qualitative and quantitative measures, that our methods produce better results than $PSNet$. In the following section, we present our simple, efficient, and formally defined color transfer technique for point clouds.

8.3 Point clouds color transfer

We are interested in transferring color style from a 3D model to another. 3D models are generally represented by meshes, a set of triangles connected through vertices. However, we decided to apply the color transfer to 3D point clouds, which is becoming a largely used standard because of its more general and compact format.

Furthermore, we assume that color transfer between 3D point clouds depends on their geometry. There are at least two features that characterize the geometry: positions and normals. We have chosen to rely on normals because of their robustness. Indeed, normals are efficient as they characterize the local shape of a surface. Besides, unlike positions, normals are invariant to scale and translation.

Before elaborating on the proposed pipeline, we formally define the term of 3D point clouds. A point cloud is a set of data points in a 3-dimensional space. A point is generally defined by its position in space (x, y, z), its color (r, g, b) and the direction of its normal to the surface (n_x, n_y, n_z).

The main goal is to transfer the color style of a target point cloud $\mathbf{T}$ to an input point cloud $\mathbf{I}$.

Let $\mathcal{P}^{\mathbf{I}}$, $\mathcal{C}^{\mathbf{I}}$ and $\mathcal{N}^{\mathbf{I}}$ be the set of positions, colors and normals vectors of the input point cloud respectively. $\mathcal{P}^{\mathbf{I}} = (\mathbf{p}^{\mathbf{I}}_1, ..., \mathbf{p}^{\mathbf{I}}_N)$, with $\mathbf{p}^{\mathbf{I}}{}_i = (x^I_i, y^I_i, z^I_i)$; $\mathcal{C}^{\mathbf{I}} = (\mathbf{c}^{\mathbf{I}}_1, ..., \mathbf{c}^{\mathbf{I}}_N)$, with $\mathbf{c}^{\mathbf{I}}_i = (r^I_i, g^I_i, b^I_i)$; and $\mathcal{N}^{\mathbf{I}} = (\mathbf{n}^{\mathbf{I}}_1, ..., \mathbf{n}^{\mathbf{I}}_N)$, with $\mathbf{n}^{\mathbf{I}}_i = (n^I_{xi}, n^I_{yi}, n^I_{zi})$ for $i \in \{1, ..., N\}$ where N is the number of points of the input point cloud.

In the same way, let $\mathcal{P}^{\mathbf{T}}$, $\mathcal{C}^{\mathbf{T}}$ and $\mathcal{N}^{\mathbf{T}}$ be respectively the set of positions, colors and normals vectors of the target point cloud. $\mathcal{P}^{\mathbf{T}} = (\mathbf{p}^{\mathbf{T}}_1, ..., \mathbf{p}^{\mathbf{T}}_M)$, with $\mathbf{p}^{\mathbf{T}}_j = (x^T_j, y^T_j, z^T_j)$; $\mathcal{C}^{\mathbf{T}} = (\mathbf{c}^{\mathbf{T}}_1, ..., \mathbf{c}^{\mathbf{T}}_M)$, with $\mathbf{c}^{\mathbf{T}}_j = (r^T_j, g^T_j, b^T_j)$; and $\mathcal{N}^{\mathbf{T}} = (\mathbf{n}^{\mathbf{T}}_1, ..., \mathbf{n}^{\mathbf{T}}_M)$, with $\mathbf{n}^{\mathbf{T}}_j = (n^T_{xj}, n^T_{yj}, n^T_{zj})$ for $j \in \{1, ..., M\}$ where M is the number of points of the target style point cloud.

Note that, the exponent T is an index standing for target rather than for transposed matrix, and the number of points in the input point cloud N is not necessary the same as the number of points in the target point cloud M.

8.3.1 Method overview

In this section, we propose two point clouds color transfer methods. As they account for the point clouds geometry, the color transferred to a point depends on the direction of its normal. Our two methods, defined in Section 8.3.3, are based on a statistical model applied to the color distributions of the point clouds.

1. our first method, named IGD_N, is a geometric approach
2. our second method, named MGD_N, relies on the correlations between color channels and normal directions

The general flowchart of our color transfer is presented in Figure 8.1 and consists of three steps.

First, we project both colors and normals vectors into new spaces where the components are less correlated, making the color transfer more robust. More precisely, for both input and target point clouds, the color (orange boxes in Figure 8.1) of each point is converted from the RGB color space to the decorrelated $l\alpha\beta$ color space [106], while the normal (blue boxes in Figure 8.1) is projected to the best fitting 3D coordinate system by Principal Component Analysis (PCA) before being decomposed into a 6-dimensional vector to separate its positive and negative components as explained in the following. A 3D coordinate system is a set of 3D basis vectors called from now on basis.

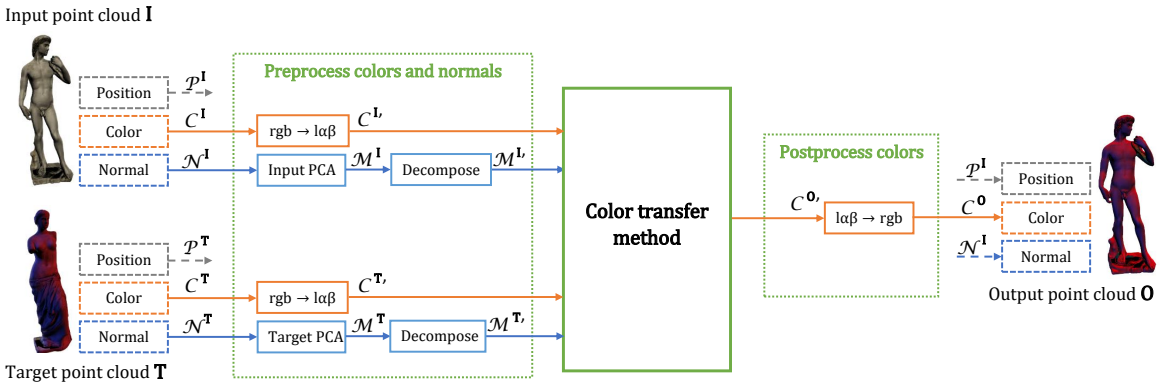


Figure 8.1: Our point clouds color transfer consists of three steps: First, normals and colors of both input **I** and target **T** point clouds are transformed. Then, a color transfer method relying on a statistical analysis of color distributions and accounting for the point clouds geometry is performed. Finally, the output color is converted back to RGB and applied to the output point cloud **O**.

Second, we apply a transformation to the input's color distribution to fit with the shape of the target's one, depending on the direction of the points normal. To do so, we propose two methods that make different assumptions regarding the shape of the distributions of the point clouds colors. The first method supposes that the color channels are independent and follow Gaussian distributions. We propose to perform six different color transfers, one for each direction of the point's normal as detailed in Section 8.3.3. The second supposes that points normals and colors follow a Multivariate Gaussian Distribution, assuming that the color channels and normal directions are correlated to each other. In that case, we give the input distribution the aspect of the target's one thanks to a linear transformation, as proposed by Pitié and Kokaram [107]. To do so, the colors and the normals of point clouds are concatenated to obtain a 9-dimensional probability distribution (3 components of color and 6 components of normal) for both input and target point clouds. A linear transformation is computed between the covariance matrices of these two 9D distributions as explained in Section 8.3.3.

Finally, regardless of the used method, we convert the colors of the output back to the RGB color space to obtain the resulting point cloud. All the steps of our pipeline are detailed in the following sections.

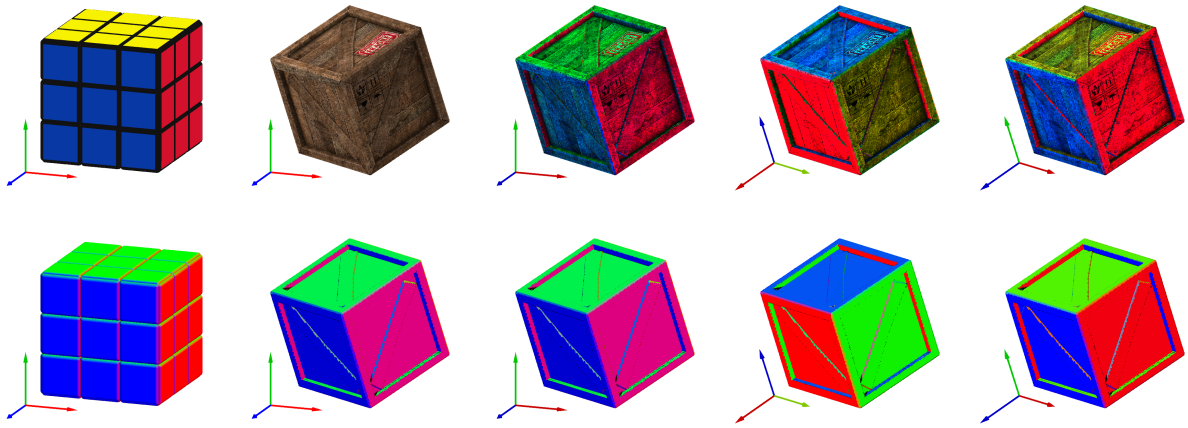
8.3.2 Making point clouds components independent

To make our color transfer more robust, we decorrelate the color channels from each other, as well as the normal directions. Regarding color, we use the perceptually decorrelated $l\alpha\beta$ color space [106]. The expression of the color space transform stems from [105]. On the other hand, the point clouds normals are not necessarily well aligned with the origin axes, i.e. a vector of normal may be oriented in between 2 or 3 axes of the basis. So, we first project the normals into the best fitting basis using a PCA, as explained in the following section. Then, the normal vectors are assigned 6 components corresponding to 6 axis directions of the basis by separating the positive from the negative directions, as detailed later.

Transforming normals basis by PCA

To make our methods more efficient in determining correlations between color channels and normal directions, we project the point normals into the best fitting basis by PCA before proceeding to the color transfer. A PCA is computed by performing an eigenvalue decomposition on the covariance matrix of the normals and used to project the normals into a new orthogonal basis that minimizes the distance from normals to origin axes. In other words, a PCA defines a more representative basis for the set of normals. Two PCA are applied: one to the input point cloud and another to the target one.

To make the aforementioned point clear and to illustrate it, Figure 8.2 shows the result of our color transfer (using our method based on a Multivariate Gaussian Distribution (MGD_N) as explained later) applied to an input point cloud that is not well aligned with the original basis, i.e. the coordinate system in which the point cloud is defined. By alignment, we mean the normals of the point cloud are aligned with the origin axes of the basis. In our example, while the target (Figure 8.2a) is aligned with its original basis, the input (Figure 8.2b) is not aligned with its original basis. However, we expect the right, up, and forward faces of the input point cloud to be colored in red, yellow, and blue respectively, following the faces' color of the target point cloud. As the input point cloud is not well aligned with the original basis, the faces' colors of the output (Figure 8.2c) result from the blending of different colors from the target point cloud (Figure 8.2a). Instead of appearing red, the right face of the output appears purple, a blending between the blue and red faces of the target, as well as the up face appears green, a blending between the yellow and blue faces of the target.



(a) Target (b) Input (c) Output (d) Output with PCA (e) Output with reoriented PCA

Figure 8.2: Red, green, and blue arrows correspond to x, y, and z axes of the basis respectively. First row: the colored point clouds. Second row: the corresponding point clouds normals (normals components x, y and z are colored in red, green, and blue respectively). The target point cloud (8.2a) is well aligned with the original basis: its normals are oriented toward the x direction for the right face (red), the y direction for the up face (green), and the z direction for the forward face (blue). As the input (8.2b) is not well aligned with the original basis (its normals are in between x, y and z axes), the color transfer fails in reproducing the colors of the target on each face of the output (8.2c). Transforming the normals by PCA (8.2d) allows a more coherent color transfer depending on the direction of the normals, but the global orientation is not preserved (axes are inverted). Automatically mapping each axis given by the PCA with the closest axis of the original basis results in the expected color transfer (8.2e).

Projecting the input normals into a more representative basis using a PCA results in well-transferred colors for each face (Figure 8.2d). Nevertheless, while the directions of the original basis are fixed (in our case: x right, y up, and z forward; respectively red, green, and blue arrows), the transformation of the normals produced by the PCA does not conserve the original global orientation (axes may be inverted).

Therefore, as the color transfer depends on the direction of the normals, the color of the target in a particular direction is transferred to the input in the same direction. For example, target points with normals oriented toward the up (y axis) are colored in yellow. This color is transferred to the points with normals oriented toward the new y axis defined by the PCA of the output (the forward direction in Figure 8.2d). Hence, the direction of the normals should match between the two point clouds. For this reason, we associate

each axis, resulting from the application of PCA, with the closest axis of the original basis using a cosine distance (Figure 8.2e). Indeed, as axes resulting from the PCA can be positively or negatively oriented, we retain the orientation which maximizes the cosine distance to an axis of the original basis (x, y, or z). As a PCA defines an orthogonal basis, the mapping ensures that each axis defined by the PCA matches exactly with one axis of the original basis. Rather than using PCA, users can also manually associate the axes, or apply a rotation to the point clouds, to obtain specific results.

In conclusion, once the normals are transformed by PCA and reoriented to match with the original basis, we decompose the normal vectors into the 6 directions of the basis to separate the positive from the negative components. This decomposition is detailed below.

Decomposing normals

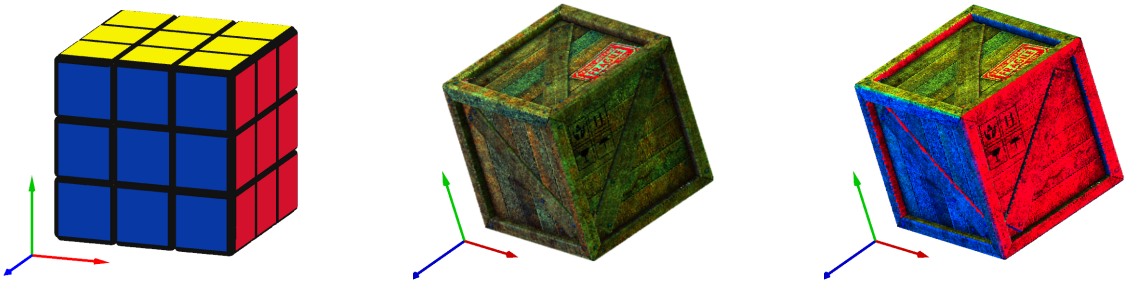
Once again, to increase the robustness of our models, we decompose the normals into positive and negative components, which results in a 6-dimensional normal vector $(-x, -y, -z, +x, +y, +z)$. This decomposition provides new sets of decomposed normal vectors for both input and target point clouds respectively: $\mathcal{M}^{\mathbf{I}\prime} = (\mathbf{m^I}_1, ..., \mathbf{m^I}_N)$ and $\mathcal{M}^{\mathbf{T}\prime} = (\mathbf{m^T}_1, ..., \mathbf{m^T}_M)$ in Figure 8.1. For an index k that can be equal to x, y or z, the components of the decomposed normal vectors are defined as follows:

$$\begin{aligned} n_{-k} &= |min(n_k, 0)| \\ n_{+k} &= |max(n_k, 0)| \end{aligned} \tag{8.1}$$

Then, the components values of input and target normals, $\mathbf{m^I}_i$ and $\mathbf{m^T}_j$ respectively, are equal to:

$$\begin{aligned} \mathbf{m^I}_i &= (n^I_{-xi}, n^I_{-yi}, n^I_{-zi}, n^I_{+xi}, n^I_{+yi}, n^I_{+zi}) \\ \mathbf{m^T}_j &= (n^T_{-xj}, n^T_{-yj}, n^T_{-zj}, n^T_{+xj}, n^T_{+yj}, n^T_{+zj}) \end{aligned} \tag{8.2}$$

Figure 8.3 visually compares the style transfer considering 3-dimensional normal vectors with the one considering decomposed 6-dimensional normal vectors (using our MGD_N method as explained later). Moreover, the correlation coefficient values from covariance matrices between colors (in RGB color space) and normals (in 3D or decomposed 6D) are given in Table 8.1. For example, the faces of the Rubik's cube oriented in the x positive and x negative directions are both composed of red color (red for $x+$ and orange for



(a) Target (b) Output with 3D normals (c) Output with decomposed 6D normals

Figure 8.3: Correlations between colors and normals are better captured when decomposing the normals into 6 directions as showed in Table 8.1. Decomposing normals results in a more coherent color transfer.

		Correlation coefficient			**Color**
		R	**G**	**B**	
Axes (3D)	**x**	-0.023	-0.160	0.080	
	y	0.072	0.054	-0.530	
	z	0.016	0.295	-0.262	
Directions (6D)	$-\mathbf{x}$	0.246	-0.112	0.283	orange
	$+\mathbf{x}$	0.159	-0.364	-0.152	red
	$-\mathbf{y}$	0.131	0.209	0.532	white
	$+\mathbf{y}$	0.246	0.297	-0.313	yellow
	$-\mathbf{z}$	-0.352	0.224	-0.101	green
	$+\mathbf{z}$	-0.377	-0.249	0.318	blue

Table 8.1: Correlation coefficient values between RGB colors and 3D normals or 6D normals for the target point cloud (Figure 8.3a). Colors in the last column result from the three RGB correlation coefficients normalized between 0 and 1. The color obtained for each direction of the 6D normals corresponds to the color of the Rubik's cube's face in this direction.

$x-$). If we consider 3D normal vectors, the correlation coefficient value between the red color and the x axis is low (-0.023), which means there is no correlation between these two components. As our methods rely on the correlation between colors and normals, considering axes with 3D normal vectors does not allow an efficient color transfer (see Figure 8.3b). On the other hand, when the normals are expressed with 6D vectors, the correlation coefficients between red color and x positive direction and x negative direction are both positive (0.246 and 0.159 respectively). When considering 6D normal vectors, the color transfer preserves the red color for both positive and negative directions of the

x axis, which results in an efficient color transfer (see Figure 8.3c and Table 8.1).

To sum up, our color transfers rely on the correlations between color terms (in $l\alpha\beta$) and geometric terms (6D normals transformed by PCA). In the following section, we detail our two methods of color transfer between point clouds.

8.3.3 Color transfer methods

Our two methods are inspired by existing color transfer techniques, applied to the geometry of point clouds, and relying on a statistical analysis of color distributions considering the normals of the point clouds. The first one assumes that the color channels are independent and follow Gaussian distributions. The second assumes that the 9-dimensional concatenated vectors of colors and normals follow a Multivariate Gaussian Distribution. The two following sections are organized as follows: First, we present the existing color transfer method that we have applied to point cloud colors. Then, we go further by improving this method by leveraging the point cloud geometry.

A. First method: Independent Gaussian Distributions

Color based Independent Gaussian Distributions. Our first method relies on the color transfer approach proposed by Reinhard *et al.* [105], called IGD (Independent Gaussian Distributions) from now on. Let us summarize this approach before presenting our first method. The goal of this IGD method is to transform the shapes of each color channel distribution of the input point cloud to match as much as possible with the corresponding target color channel distribution. Assuming that the color channels are independent, Reinhard *et al.* suppose that each color channel follows a Gaussian distribution. For this purpose, the mean μ and the standard deviation σ of each component of the $l\alpha\beta$ channels are computed, for both input and target point clouds. Then, the resulting output color channels are computed as follows:

$$\begin{aligned} l_i^O &= (l_i^I - \mu_l^I) \times \frac{\sigma_l^T}{\sigma_l^I} + \mu_l^T \\ \alpha_i^O &= (\alpha_i^I - \mu_\alpha^I) \times \frac{\sigma_\alpha^T}{\sigma_\alpha^I} + \mu_\alpha^T \\ \beta_i^O &= (\beta_i^I - \mu_\beta^I) \times \frac{\sigma_\beta^T}{\sigma_\beta^I} + \mu_\beta^T \end{aligned} \tag{8.3}$$

Finally, we defined the output color $\mathbf{c^O}_i{}' = (l_i^O, \alpha_i^O, \beta_i^O)$. At this stage, the color transfer applied to the point clouds does not account for geometry.

Color and geometry based Independent Gaussian Distributions. Hereafter, we describe our first method, called from now on IGD_N. We perform the color transfer depending on both color and geometry. To do so, we compute the means and the standard deviations of each of the three $l\alpha\beta$ component weighted by the six components of the decomposed normal vector ($\mathbf{m^I}$ and $\mathbf{m^T}$ for the input and target point cloud defined in Equation 8.2). It results in 18 means μ_{kc} and 18 standard deviations σ_{kc} for both the input and target point clouds, where the index c corresponds to one of the three color component $l\alpha\beta$, and the index k corresponds to one of the six component of the normal $(-x, -y, -z, +x, +y, +z)$:

$$\begin{aligned} \mu_{kc}^I &= \frac{1}{\sum_{i=0}^{N} n_{ki}^I} \sum_{i=0}^{N} n_{ki}^I \cdot c_i^I \\ \sigma_{kc}^I &= \frac{1}{\sum_{i=0}^{N} n_{ki}^I} \sum_{i=0}^{N} n_{ki}^I \cdot \left(c_i^I - \mu_{kc}^I\right)^2 \end{aligned} \tag{8.4}$$

$$\begin{aligned} \mu_{kc}^T &= \frac{1}{\sum_{j=0}^{M} n_{kj}^T} \sum_{j=0}^{M} n_{kj}^T \cdot c_j^T \\ \sigma_{kc}^T &= \frac{1}{\sum_{j=0}^{M} n_{kj}^T} \sum_{j=0}^{M} n_{kj}^T \cdot \left(c_j^T - \mu_{kc}^T\right)^2 \end{aligned} \tag{8.5}$$

Once the 18 μ_{kc} and 18 σ_{kc} are computed for both the input and target point clouds, we perform the three color transfers of each input point for the six directions using Equation 8.3, resulting in six new $l_{ki}^O, \alpha_{ki}^O, \beta_{ki}^O$ colors per point:

$$\begin{aligned} \boldsymbol{l}_i^O &= \left(l_{-xi}^O, l_{-yi}^O, l_{-zi}^O, l_{+xi}^O, l_{+yi}^O, l_{+zi}^O\right) \\ \boldsymbol{\alpha}_i^O &= \left(\alpha_{-xi}^O, \alpha_{-yi}^O, \alpha_{-zi}^O, \alpha_{+xi}^O, \alpha_{+yi}^O, \alpha_{+zi}^O\right) \\ \boldsymbol{\beta}_i^O &= \left(\beta_{-xi}^O, \beta_{-yi}^O, \beta_{-zi}^O, \beta_{+xi}^O, \beta_{+yi}^O, \beta_{+zi}^O\right) \end{aligned} \tag{8.6}$$

Finally, the output $l_i^O, \alpha_i^O, \beta_i^O$ colors result from a linear combination between the color components and the values of the 6D normal vector:

$$\begin{aligned} l_i^O &= < \mathbf{m^I}_i, \boldsymbol{l}_i^O > \\ \alpha_i^O &= < \mathbf{m^I}_i, \boldsymbol{\alpha}_i^O > \\ \beta_i^O &= < \mathbf{m^I}_i, \boldsymbol{\beta}_i^O > \end{aligned} \tag{8.7}$$

This scalar product between normals and color components ensures a smooth color transition for points with a normal direction in-between 2 or 3 axes. Finally, we define the output color as $\mathbf{c^O}_i{}' = (l_i^O, \alpha_i^O, \beta_i^O)$.

B. Second method: Multivariate Gaussian Distribution

Color based Multivariate Gaussian Distribution. Our second method relies on the color transfer approach proposed by Pitié and Kokaram [107], called *MGD* (Multivariate Gaussian Distribution) from now on. Let us summarize this approach before presenting our second method. The goal of this transformation is to match an input MGD (Multivariate Gaussian Distribution) with a target MGD using the Monge-Kantorovich closed-form mapping. Without considering the geometry, given the means (μ) and covariances ($\sum$) of input and target $l\alpha\beta$ color distributions, the transformation matrix of Monge-Kantorovich mapping is computed as:

$$\mathbf{M} = \sum_{\mathcal{C}^{\mathbf{I}\prime}}^{-1/2} \left(\sum_{\mathcal{C}^{\mathbf{I}\prime}}^{1/2} \sum_{\mathcal{C}^{\mathbf{T}\prime}} \sum_{\mathcal{C}^{\mathbf{I}\prime}}^{1/2} \right)^{1/2} \sum_{\mathcal{C}^{\mathbf{I}\prime}}^{-1/2} \tag{8.8}$$

with $\mathcal{C}^{\mathbf{I}\prime}$ and $\mathcal{C}^{\mathbf{T}\prime}$ the sets of $l\alpha\beta$ color vectors for the input and target point clouds respectively, as defined in Section 8.3. Finally, the new set of color vectors for the output point cloud $\mathcal{C}^{\mathbf{O}\prime} = (\mathbf{c^O}_1{}', ..., \mathbf{c^O}_N{}')$ are computed as:

$$\mathbf{c^O}_i{}' = (\mathbf{c^I}_i{}' - \boldsymbol{\mu}_{\mathcal{C}^{\mathbf{I}\prime}}) \cdot \mathbf{M} + \boldsymbol{\mu}_{\mathcal{C}^{\mathbf{T}\prime}}, \qquad i \in \{1, ..., N\}, \tag{8.9}$$

with $\mathbf{c^I}_i{}'$, $\boldsymbol{\mu}_{\mathcal{C}^{\mathbf{I}\prime}}$ and $\boldsymbol{\mu}_{\mathcal{C}^{\mathbf{T}\prime}}$ are the input color in $l\alpha\beta$, the mean color of the input and the mean color of the target respectively.

Color and geometry based Multivariate Gaussian Distribution. Let us describe our second method, called from now on MGD_N. In order to consider the point clouds

geometry, we extend the MGD method by concatenating colors and normals vectors to obtain sets of 9-dimensional vectors $f^{\mathbf{I}}$ and $f^{\mathbf{T}}$ for the input and target point clouds respectively where:

$$f_i^{\mathbf{I}} = (l_i^I, \alpha_i^I, \beta_i^I, n_{-xi}^I, n_{-yi}^I, n_{-zi}^I, n_{+xi}^I, n_{+yi}^I, n_{+zi}^I), \tag{8.10}$$

$$f_j^{\mathbf{T}} = (l_j^T, \alpha_j^T, \beta_j^T, n_{-xj}^T, n_{-yj}^T, n_{-zj}^T, n_{+xj}^T, n_{+yj}^T, n_{+zj}^T). \tag{8.11}$$

Given the means $\boldsymbol{\mu}_{f^{\mathbf{I}}}$, $\boldsymbol{\mu}_{f^{\mathbf{T}}}$ and the covariance matrices $\sum_{f^{\mathbf{I}}}$, $\sum_{f^{\mathbf{T}}}$ of the input and target distributions $f^{\mathbf{I}}$ and $f^{\mathbf{T}}$ respectively, the transformation of the input distribution is given by:

$$\mathbf{M} = \sum_{f^{\mathbf{I}}}^{-1/2} \left(\sum_{f^{\mathbf{I}}}^{1/2} \sum_{f^{\mathbf{T}}} \sum_{f^{\mathbf{I}}}^{1/2} \right)^{1/2} \sum_{f^{\mathbf{I}}}^{-1/2}, \tag{8.12}$$

and the samples of the output distribution are computed as:

$$f_i^{\mathbf{O}} = \left(f_i^{\mathbf{I}} - \boldsymbol{\mu}_{f^{\mathbf{I}}} \right) \cdot \mathbf{M} + \boldsymbol{\mu}_{f^{\mathbf{T}}}, \qquad i \in \{1, ..., N\} \tag{8.13}$$

where $f^{\mathbf{O}}$ is the set of 9-dimensional output vectors. From this set of 9-dimensional output vectors, each vector $f_i^{\mathbf{O}}$ is composed of 3 color components and 6 normal components. By preserving only the 3 color components, we obtain the new set of output colors vectors $\mathcal{C}^{\mathbf{O}\prime} = (\mathbf{c^O}_1{}', ..., \mathbf{c^O}_N{}')$, where each $\mathbf{c^O}_i{}' = (l_i^O, \alpha_i^O, \beta_i^O)$.

8.3.4 Applying output colors to point cloud

The color transfer is finalized by applying the new color to the input point cloud. In the final step, the $l\alpha\beta$ output colors are converted back to the RGB colors following [105].

To sum up, each of our two methods consists of three steps:

1. The colors and normals of both the input and target point clouds are projected into more decorrelated spaces ($l\alpha\beta$ for color and PCA + decomposition for normal)
2. The color transfer is performed between the input and target using one of our proposed method (IGD_N or MGD_N)
3. The output colors of the points are converted back to the RGB color space

8.4 Results

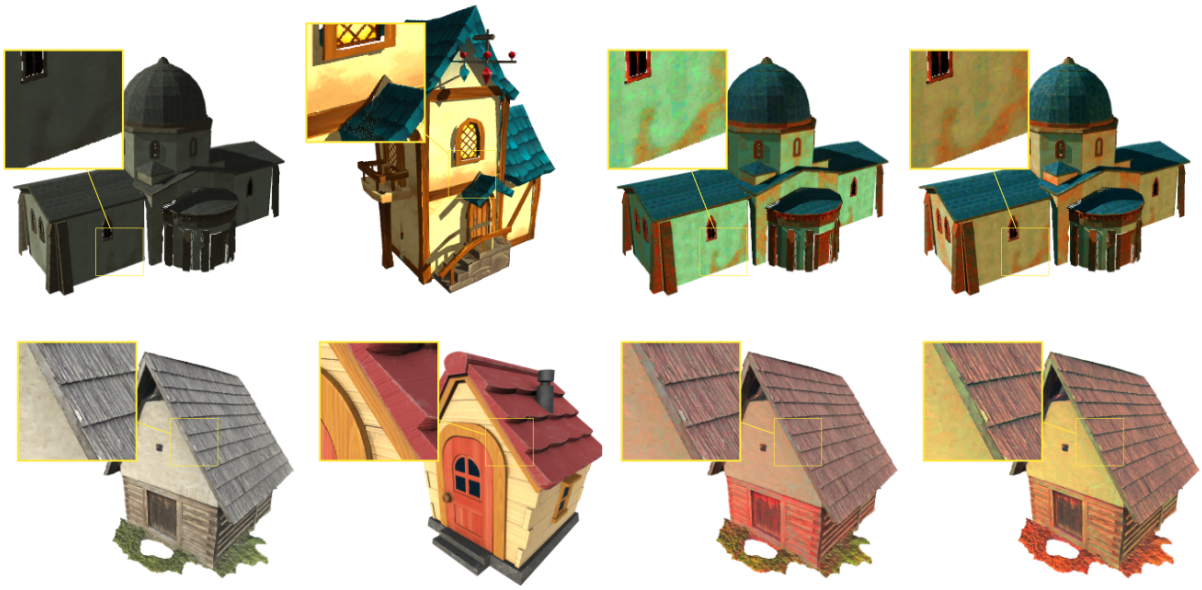
8.4.1 Qualitative comparison

In the following, we present several examples of point cloud color transfers performed by our two methods. Our proposed color transfers for point clouds can be used for building colorization as illustrated in Figure 8.4. In the first row, the color style of the fantasy house point cloud is transferred to the input church point cloud. In addition to changing the color of roofs and walls, our methods also change the lighting to fit better with the target color style. Even color variations on flat surfaces are transferred (yellow insets). In this example, MGD_N better reproduces the colors of the target. The performance of our MGD_N method regarding building colorization is confirmed by the example of the second row.

Another application is the color transfer between two inner rooms (see Figure 8.5). While our two methods perform well in transferring the colors of the king's room to the red saloon, the result produced with the IGD_N method is a bit more reddish than the result produced with the MGD_N method. In a second example, presented in Figure 8.6, the color of a room with the style of Van Gogh is transferred to a scan of a painted kitchen. The difference between the wood color of the floor and the blue color of the walls is well preserved in the resulting point clouds. Once again, the global color seems more coherent when using our MGD_N method.

The color transfer can also be done for exterior environments as illustrated in Figure 8.7. It can be useful where shooting a movie in a particular environment (the Mar Saba monastery in Cisjordan for example) and wanting to give it the color style of a different region (the Momoyama castle in Japan here). In this case, the result produced with our MGD_N method has more color variations, which better corresponds to the color style of the target. Another example is showed in Figure 8.8. Both input and target point clouds are scans of castles. Color variation between walls, path, and grass are well transferred. Even the shadows are smoothed to better match the target color style.

Finally, we used our methods for color transfer between furniture, like chairs, as illustrated in Figure 8.9. For the example of the first row, while the colors globally correspond to the target color style for both of our methods, the different parts of the chairs are not well separated. As a result, we expect the color of the chair legs to be whiter and the color of the leather to have pink nuances. For the second example (second row), the result produced when using IGD_N encounters the same problem as in the previous example.



Input Target IGD_N MGD_N

Figure 8.4: Color transfer for building colorization. Our MGD_N method better transfers the colors of the target point clouds.

Input Target

IGD_N MGD_N

Figure 8.5: Color transfer between inner rooms. IGD_N produces a slightly reddish result while MGD_N produces a faithful color transfer between the two rooms.

Figure 8.6: The input point cloud is a scan of a painted kitchen and the target is a bedroom with the color style of Van Gogh. The wood color of the floor and the blue color of the walls are well transferred with our two methods.

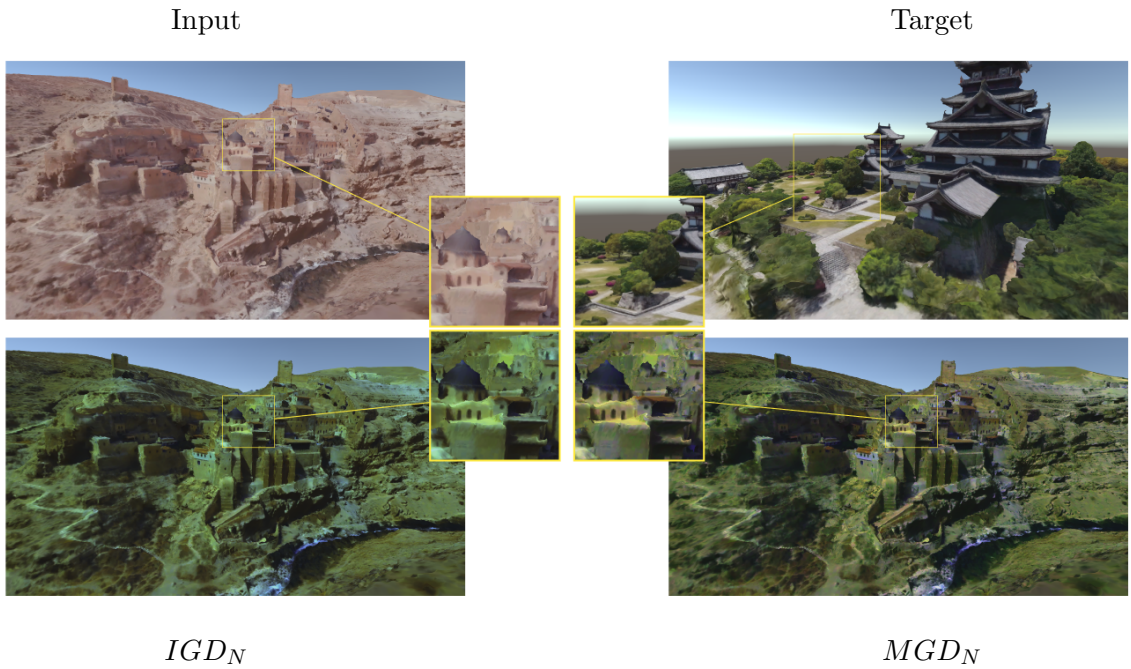


Figure 8.7: Color transfer between exterior environments. The color of the Momoyama castle in Japan is transferred to the Mar Saba monastery in Cisjordan.

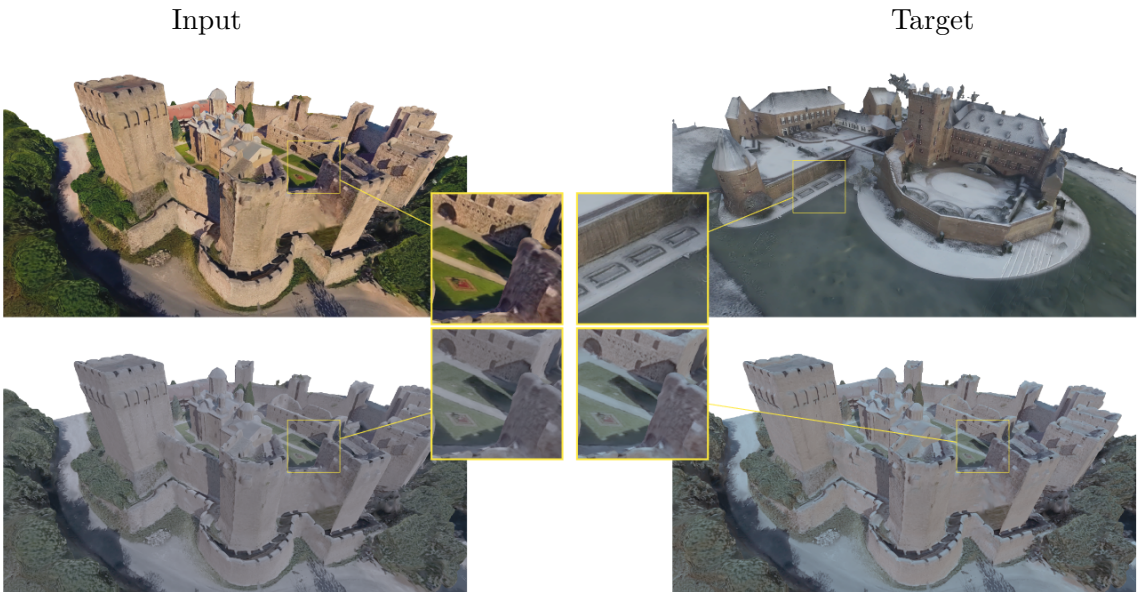


Figure 8.8: The point clouds are scans of castles. While the input shows sunny weather, the target is a scan of a castle in Norway, with snowy weather. Our two methods succeed in reproducing the snowy weather.

Figure 8.9: Color transfer between chairs. While the transferred colors globally correspond to those of the target, the different parts of the chairs are not well separated.

The color of the bamboo of the target beach chair is transferred to the fabric of the input chair. When using MGD_N, the colors are better separated and the fabric is composed of blue and white, as expected.

To conclude, both proposed methods produce qualitative color transfer depending on the considered point clouds. However, it is difficult to know why one of the methods performed better than the other in some cases. For the sake of completeness, we evaluated several techniques for transferring color between point clouds.

8.4.2 Quantitative comparison with other methods

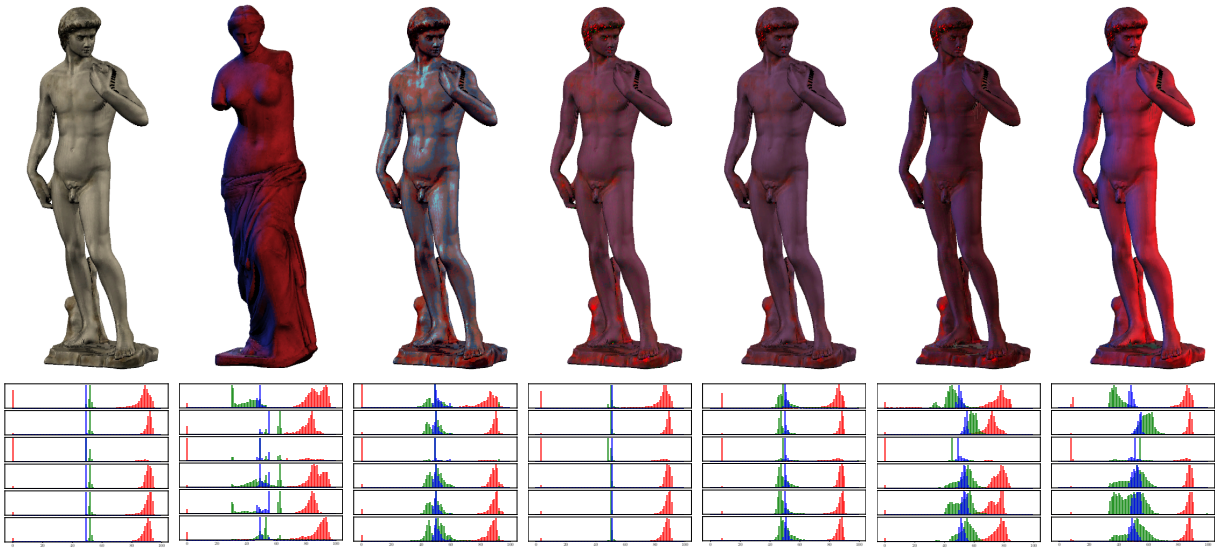
In this section, we evaluate the quality of the color transfer performed by the methods listed below:

- $PSNet$: CNN-based method proposed by *Cao et al.* [103]
- IGD: Independent Gaussian Distributions not accounting for normals (based on Reinhard *et al.* [105])
- MGD: Multivariate Gaussian Distribution not accounting for normals (based on Pitié and Kokaram [107])
- IGD_N: Independent Gaussian Distributions accounting for normals (our first method)
- MGD_N: Multivariate Gaussian Distribution accounting for normals (our second method).

Figure 8.10 illustrates a resulting color transfer for each method. The color distributions in $l\alpha\beta$ are displayed for the six directions of the basis.

For quantitatively evaluating and benchmarking the different methods, we propose several metrics that compare the color distributions of the target point cloud with the color distributions of the output point clouds. We expect the color distribution of the output to be close to the color distribution of the target. To measure the similarity between these two color distributions, we adapt objective metrics used in the field of image color transfer. The value provided by a metric will be called score from now on.

We propose three different approaches to calculate the score of similarity between the color distributions of two point clouds. First, the similarity score is calculated on the global color distribution of the point clouds (i.e. all the points of the cloud are considered), as it is done for images. Second, several viewpoint-dependent similarity scores are calculated on subsets of the points, the final similarity score is then the average of all the viewpoint scores as detailed later. A third similarity score can be computed in the case of color transfer for relighting and delighting. The expected result (i.e. the ground truth) can



(a) Input (b) Target (c) $PSNet$ (d) IGD (e) MGD (f) IGD_N (g) MGD_N

Figure 8.10: Point clouds color transfer evaluation. The second row shows the $l\alpha\beta$ distributions of the point clouds for each of the 6 directions, from top to bottom: $-x, +x, -y, +y, -z, +z$. The red, green, and blue colors correspond to l, α, and β distributions respectively.

be simulated, so the color of each output point can be compared with the color of the corresponding point of the ground truth, as explained below. In the following, as the evaluated point clouds are well aligned with their original basis, making use of a PCA was not relevant.

Global color distributions similarity score

To compute the global similarity score between the output and the target color distributions, we use the metrics defined for image color transfer. As an example, after applying for their style-aware color transfer, Hristova *et al.* [108] computed the Bhattacharya coefficient to measure the similarity between the target and the output color distributions. To evaluate the color transfer in terms of color and luminance, they calculate the score for each component of the CIELAB color space. The final similarity score is obtained by averaging the similarity scores of the luminance channel and the two chroma channels.

Besides, several works performed a color transfer based on the Wasserstein distance, between the color distributions of two images, and obtained convincing results [120], [121]. The Wasserstein distance is a solution to the Earth Mover's Distance (EMD) problem

and is computed between the output and the target color distributions. We decided to evaluate both Bhattacharya and Wasserstein distances between output and target point clouds color distributions in CIELAB color space. The final similarity score is the average of the three components scores. Both metrics give a similarity score between 0 and 1, where 1 means identical distributions (for clarity we take 1 - Wasserstein coefficient).

Overall, for 14 pairs of input and target point clouds we have evaluated, when comparing the color transfer produced by our methods against the others, our methods perform better for both Bhattacharya and Wasserstein distances as showed in Table 8.2. A Repeated Measures ANOVA has been performed and a significant difference ($p < 0.05$) between the scores obtained by the different methods has been found for both metrics. A posthoc test showed that the scores of MGD are significantly lower than those of IGD_N and MGD_N ($p < 0.05$) for the Bhattacharya metric. Regarding the Wasserstein metric, a posthoc test showed that the scores of $PSNet$ and IGD are significantly lower than those of IGD_N and MGD_N ($p < 0.05$).

Nevertheless, those metrics suffer from a limitation when it comes to comparing color distributions of point clouds as they globally consider all points of the clouds, without accounting for geometry. For example, regarding the color transfer presented in Figure 8.10, the score of $PSNet$ is higher than the scores of our two methods for the Bhattacharya metric ($PSNet$=0.943, IGD_N=0.930 and MGD_N=0.931). While the visual quality of the resulting point clouds clearly shows that our two methods (Figures 8.10f and 8.10g) perform better than $PSNet$ (Figure 8.10c), the color distribution of the target point cloud is globally closer to the color distribution of $PSNet$ than the color distributions of our two methods.

To tackle this issue, we propose to calculate similarity scores between rendered images of the two point clouds when considering many viewpoints as explained in the following section. Proceeding this way, we ensure the score to be coherent with the color transfer

	$PSNet$	IGD	MGD	IGD_N	MGD_N
Bhattacharya	0.84 ±0.136	0.87 ±0.103	0.86 ±0.105	**0.89** ±0.080	**0.90** ±0.076
Wasserstein	0.43 ±0.146	0.45 ±0.129	0.48 ±0.174	**0.50** ±0.127	**0.52** ±0.126

Table 8.2: Means and standard deviations of scores obtained with the 5 evaluated methods, for the metrics applied globally to all points of the point clouds, and for the color transfer of 14 different point clouds.

as it depends on the point clouds geometry visualized from a particular point of view.

Per viewpoint color distributions similarity score

To improve the relevance of the comparison, we calculate several similarity scores depending on the viewpoint the point clouds are visualized from. The point clouds are rendered using splatting techniques with a point radius large enough to ensure no holes in the projection of the 3D models and the background is not considered in the color distribution. Moreover, the images are rendered from viewpoints uniformly distributed on a sphere surrounding the point clouds. Following the suggestion of Alexiou *et al.* [122] in their work that exploits user interactivity to assess point clouds quality, the cameras are uniformly placed on the vertices of a geodesic sphere as illustrated in Figure 8.11. This placement uniformly considers all the point cloud areas when computing the average scores. We decided to render 642 views per point cloud to obtain a dense covering, where each rendered image is 720×720 pixels.

The Bhattacharya and Wasserstein scores are computed on the color distributions of each rendered view of the target and output point clouds. The resulting scores show that the color distributions of the two point clouds are similar when visualized from a particular viewpoint. Moreover, we can display these scores on a heat-map projected onto the sphere surrounding the point clouds as showed in Figure 8.12. Each point of the sphere corresponds to a viewpoint, the color of a point corresponds to the similarity score (Wasserstein in this example) where blue means not similar and red means similar. We see that, with our two methods (Figures 8.12d and 8.12e), globally, the similarity scores are

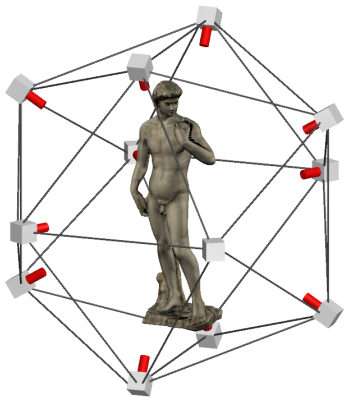
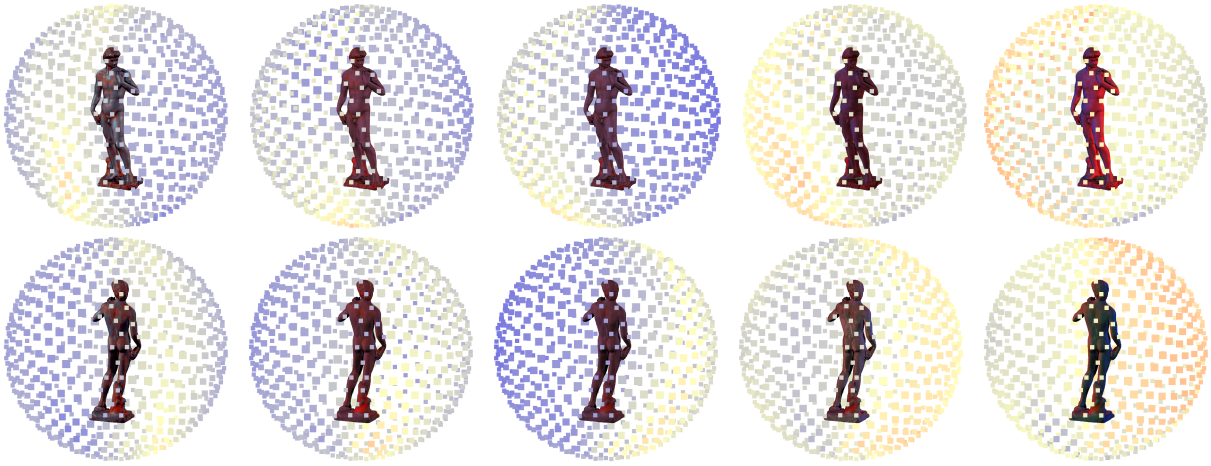


Figure 8.11: Cameras are uniformly placed on the geodesic sphere surrounding the point cloud.



(a) $PSNet$	(b) IGD	(c) MGD	(d) IGD_N	(e) MGD_N
min=0.271	min=0.294	min=0.231	min=**0.361**	min=0.349
max=0.561	max=0.596	max=0.553	max=0.608	max=**0.651**
mean=0.378	mean=0.380	mean=0.353	mean=0.461	mean=**0.499**
std=0.053	std=0.055	std=0.081	std=0.053	std=0.063

Figure 8.12: The two lines represent the same point clouds visualized from two different viewpoints. Each point of the surrounding sphere corresponds to a viewpoint. The color of the point depends on the Wasserstein score in this example. Blue color means not similar (low score) and red color means similar (high score). The color distributions of our methods (8.12d and 8.12e) are globally more similar to the target than the other methods.

higher than those obtained with other methods (i.e. the surrounding spheres are redder everywhere).

Finally, a unique score is computed by averaging all the viewpoint-dependent similarity scores. This unique score corresponds to how the color distributions are similar when visualizing the point clouds from every viewpoint, on average. Once again, our methods obtain the best overall scores for both Bhattacharya and Wasserstein metrics, as shown in Table 8.3. A Repeated Measures ANOVA still shows a significant difference ($p < 0.05$) between the scores obtained with the different methods for both metrics. A posthoc test showed that the scores of $PSNet$ are significantly lower than those of IGD_N and MGD_N ($p < 0.05$), scores of IGD are significantly lower than those of IGD_N, and scores of MGD are significantly lower than those of MGD_N for the Bhattacharya metric. Regarding the Wasserstein metric, a posthoc test showed the scores of $PSNet$ to be significantly lower than those of MGD, IGD_N, and MGD_N ($p < 0.05$), and scores of IGD to be lower than those of IGD_N and MGD_N.

	$PSNet$	IGD	MGD	IGD_N	MGD_N
Battacharya	0.81 ±0.147	0.84 ±0.115	0.84 ±0.112	**0.87** ±0.096	**0.88** ±0.081
Wasserstein	0.41 ±0.145	0.44 ±0.136	0.46 ±0.165	**0.47** ±0.122	**0.50** ±0.118

Table 8.3: Means and standard deviations of scores obtained by the 5 evaluated methods, for the averaged viewpoint dependent metrics, and for the color transfer of 14 different point clouds.

This viewpoint-dependent metric is then more coherent than the score computed over the global color distributions. Using the same example as previously (Figure 8.10), the Bhattacharya score of our two methods is now higher than the score of $PSNet$ ($PSNet$=0.874, IGD_N=0.894 and MGD_N=0.882). In this case, the viewpoint-dependent metric is more representative of the color transfer quality than the global metric presented earlier.

Figure 8.13 shows an example where Bhattacharya and Wasserstein viewpoint dependent scores are representative of the visual quality of the color transfer. Indeed, with MGD_N the orange color of the target flowers is not preserved and the variety of green colors are not rendered on the leaves. Contrarily, IGD_N provides a faithful and coherent color transfer.

Point-to-point distance with ground truth for relighting and delighting use case

We finally propose a third metric that can only be used in the case of relighting and delighting. Relighting is the process of applying the lighting condition of a target point cloud to an input one. On the opposite, a delighting process is a high trend for a few years and consists of removing the lighting of a model. In this scenario, the ground truth can be simulated by rendering the input point cloud with the lighting conditions of the target point cloud as illustrated in Figure 8.14. The output point cloud is then compared with the simulation (i.e. the ground truth). As a result, each point of the input cloud is assigned two colors: one resulting from the color transfer and the other resulting from the relighting (the ground truth). We compute the PSNR and the cosine distance applied to the pairs of colors (output and ground truth). Note that the two scores (PSNR and cosine distance) are computed for each color component in CIELAB. The L, a, and b scores are then averaged to obtain the final scores.

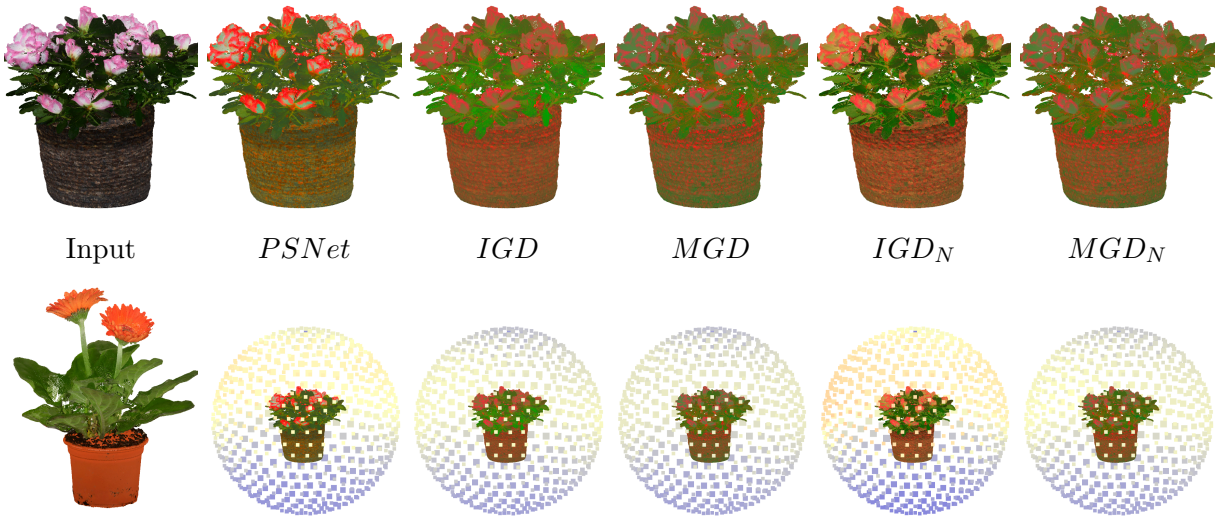


Figure 8.13: An example where MGD_N fails in providing a qualitative color transfer while IGD_N provides a faithful and coherent color transfer. The surrounding spheres are colored with the Wasserstein scores while the means of the Bhattacharya (B) and the Wasserstein (W) viewpoint dependent scores are reported in the caption. Both scores are coherent with the visual quality of the results.

Figure 8.14: Simulating the expected result of relighting.

Figure 8.15 illustrates two examples of relighting (first row) and delighting (second row). For those examples, the cosine distance is more representative of the visual quality of the color transfer than the PSNR. All the metrics presented in this section (global, per viewpoint, and point-to-point similarity scores) are valid for color transfer used for relighting or delighting applications.

To conclude, according to the objective metrics, we show that accounting for the point

Ground truth	*PSNet*	*IGD*	*MGD*	IGD_N	MGD_N
	PSNR=11.511	PSNR=17.486	PSNR=18.637	PSNR=**19.084**	PSNR=16.938
	C=0.937	C=0.954	C=0.956	C=0.978	C=**0.986**

Ground truth	*PSNet*	*IGD*	*MGD*	IGD_N	MGD_N
	PSNR=12.272	PSNR=11.283	PSNR=12.520	PSNR=13.793	PSNR=**13.866**
	C=0.989	C=0.994	C=0.985	C=**0.996**	C=0.994

Figure 8.15: The scores of the point-to-point metrics for relighting and delighting. The cosine distance (C) is coherent with the visual quality of the results while the PSNR is not very representative.

cloud geometry improves significantly the color transfer, and even more when considering correlations between colors and normals. In the next section, we present few other applications that can be tackled by our color transfer methods for point clouds.

8.4.3 Other applications

While, with the presented results below, the input and target point clouds have similar geometric structures, we proposed few other applications where the geometric structures are not necessarily similar between the two point clouds. The first ideas of applications are concerned with aging and weathering. Several techniques exist for those purposes [123]. Aging consists in gradually deteriorating a model depending on a period or on a chosen color style as illustrated in Figure 8.16 (MGD_N method has been used to produce these results). Here, we introduce the possibility of applying the color style to the output depending on a coefficient α:

$$\mathbf{new}^{\mathbf{O}}_{\mathbf{c}\ i} = \alpha \times \mathbf{c}^{\mathbf{O}}{}_i + (1-\alpha) \times \mathbf{c}^{\mathbf{I}}{}_i \tag{8.14}$$

where $\mathbf{new}^{\mathbf{O}}_{\mathbf{c}\ i}$ is a linear combination between the input color $\mathbf{c}^{\mathbf{I}}{}_i$ and the output color $\mathbf{c}^{\mathbf{O}}{}_i$ of the point i, which replaces the output color in the final rendering. In this example, the statue is more or less aged thanks to the target color style of a mossy stump.

On the other hand, weathering refers to embedding a model at a particular season time with specific environmental conditions. In the example presented in Figure 8.17, we

Target | Input ($\alpha = 0$) | Output ($\alpha = 0.25$) | Output ($\alpha = 0.5$) | Output ($\alpha = 0.75$) | Output ($\alpha = 1$)

Figure 8.16: Gradually aging using MGD_N color transfer, with α a coefficient of linear combination between input and output point clouds. The input point cloud becomes more and more greenish on its exterior surfaces.

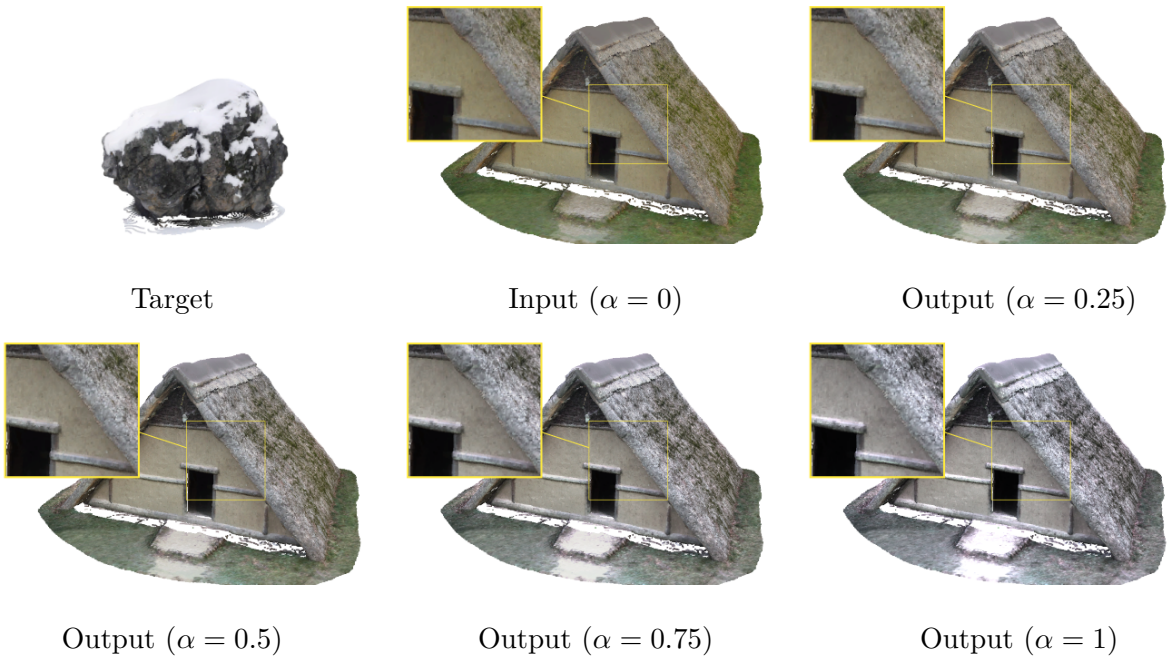


Figure 8.17: Gradually weathering using MGD_N color transfer, with α a coefficient of linear combination between input and output point clouds. The input point cloud becomes more and more snow-covered on its surfaces oriented toward the up direction.

embed an archaic house in a snowy environment defined by a simple snowed rock as a target color style. The strong point of our method is to transfer color depending on the direction of the normals. In the target point cloud, the normals oriented toward the up direction correspond to snow and are white. In the outputs, the surfaces oriented toward the up direction are then whiter, corresponding to snow that better holds (see the door trim in the yellow insets).

Finally, as it has been presented in Section 8.4.2, relighting and delighting are two applications of our color transfer for point clouds. Our method is very efficient for this task as lighting directly depends on normals, as showed in Figure 8.18. In this example, IGD_N does not succeed in transferring the red color on the right side of the statue while MGD_N produces a color transfer very close to the expected result.

Furthermore, our method allows delighting point clouds by taking a neutral point cloud as a target color style, as illustrated in Figure 8.19. Contrary to the previous example of relighting, IGD_N color transfer results in a uniform grey color on the entire statue while color variations are still perceptible on the MGD_N resulting point cloud.

Input Target IGD_N MGD_N

Figure 8.18: Point clouds color transfer used for relighting.

Input Target IGD_N MGD_N

Figure 8.19: Point clouds color transfer used for delighting.

8.5 Conclusion

Point clouds are becoming a commonly used data format for many applications, such as cultural heritage preservation, scanning of the surrounding environment, or the generation of 3D assets for video games and computer graphics movies, to name a very few. Improving the rendering quality of such point clouds is essential and has been studied for years now.

In this chapter, we have presented a new and innovative pipeline for facilitating the transfer of point clouds color. The proposed methods, which rely on the color distribution as well as the geometry, are simple, unsupervised, and efficient. Our color transfers provide very good results in terms of visual quality. Moreover, the proposed methods outperform the methods that do not account for the point clouds geometry as well as the *PSNet* method [103], in terms of visual quality and objective metrics.

Part V

Conclusion

Chapter 9

Conclusion

9.1 Summary of contributions

In this thesis, we have focused on improving the rendering quality of the new HDR 3D point cloud video format. The principal issue in this field was the tone mapping of HDR content for visualization on HMD. We first studied the human eye perception on such HMD to adapt existing CAM to the encountered visualization conditions. It allows us to tackle the tone mapping problem for HMD by fixing some constraints on the content: static 360° HDR images visualized from a fixed viewpoint. We proposed our HMD-TMO that improves the rendering quality of HDR images visualized on HMD by preserving the global coherency of the scene and enhancing the contrast in the viewport. We then relaxed the constraints by tackling the problem of tone mapping a rendered HDR image when walking through interactive 3D scenes. We proposed our 3D-TMO that introduces the definition of scene luminance and the environment maps solution we used to obtain a real-time estimation of the dynamic range of a scene. With the benefit of hindsight, our 3D-TMO is a generalization of our HMD-TMO. Tone mapping a 360° HDR image can be performed by defining this image as the unique environment map of the scene and then applied 3D-TMO from a fixed viewpoint. Thereafter, once a solution to produce a coherent and qualitative rendering of an HDR 3D point cloud on SDR screens and especially on HMD has been found, we explored the field of a stylized rendering of point clouds. Relying on the color distribution and the geometry of point clouds, we proposed simple and efficient methods to perform an example-based color transfer between point clouds. We also proposed several objective metrics to measure the quality of the color transfer. Finally, our work contributes to the field of point cloud rendering and brings answers to some questions of our research field, but more importantly, they open an avenue for improvement as discussed in the following.

9.2 Future work

In what follows, future work for each of our contributions is presented.

Lightness and color perception on Head Mounted Display

Further experiments could lead to a more accurate HMDCAM (Color Appearance Model for HMD) as the current one does not fit perfectly with the collected data. Besides, an evaluation of all HMDs with their features (resolution, dynamic range, etc.) should be performed to adapt HMDCAM to the used HMD. A subjective evaluation of the hue perception could also improve the model. We believe that whatever the type of used display (HDR, 3D, HMD, etc.), the quality of images could be improved by considering screen-dependent CAMs obtained thanks to subjective evaluations conducted on all available screens.

Tone Mapping Operator for High Dynamic Range 3D point clouds

The main issue of our 3D-TMO is inconsistency when sudden changes appear in a distant room: the global luminance changes and causes an undesired evolution of the tone mapped image. This issue is managed thanks to smooth transitions but could be tackled differently considering visual adaptation or maladaptation. Future work could head toward fixing this incoherent result. Moreover, the global and the viewport TMOs used in our 3D-TMO are not well adapted to the perception on HMD as they do not use our HMDCAM. Unfortunately, the perceptually coherent global TMO used in HMD-TMO could not be computed in real-time. Other TMOs that consider our HMDCAM and that can be computed in real-time could be used. Another improvement could be a semi or fully automated placement of the 360° cameras thanks to an analysis of the scene topology. Finally, more subjective evaluations should be conducted for manifold 3D scenes with various topologies, lighting design, and more interactivity. Evaluating our TMO on SDR 2D screens to confirm its efficiency compared with other TMOs could be of great interest.

Example-based color transfer for 3D point clouds

We believe that relaxing the Gaussian hypothesis on color distribution could improve the quality of the color transfer. For example, a mixture of multivariate generalized Gaussian distribution or a beta distribution could be good candidates for further improving

the results [109], [110]. We also aim to investigate the potential of deep networks to automatically generate stylized point clouds by manipulating several visual features, such as color, gradient, texture, and so on. More representative objective quality metrics are also required to precisely determine the quality of the transfer. Finally, a color transfer between HDR point clouds is an interesting challenge. Besides the color, the dynamic range has to be transferred, preserving details and contrasts

9.3 Discussion

Regarding the ReVeRY project, our contributions allowed finding solutions for obtaining a qualitative rendering of an HDR 3D point cloud in real-time on SDR displays and HMDs. Moreover, our color transfer method for point clouds allows more stylization of the captured content. To go further, our work on tone mapping and color transfer of 3D contents have opened avenues for improving the rendering quality of future movies and video games. However, it is hard to believe that, in a near future, HDR point clouds will be streamed to consumers as free-viewpoint high-quality content. Nevertheless, we are convinced that capturing HDR point clouds allows more flexibility and creativity to content creators for improving the quality of their images.

BIBLIOGRAPHY

[1] T. Mertens, J. Kautz, and F. Van Reeth, « Exposure fusion: A simple and practical alternative to high dynamic range photography », *in Computer graphics forum*, Wiley Online Library, vol. 28, 2009, pp. 161–171.

[2] N. Yastikli, « Documentation of cultural heritage using digital photogrammetry and laser scanning », *Journal of Cultural Heritage*, vol. 8, *4*, pp. 423–427, 2007.

[3] M. Pieraccini, G. Guidi, and C. Atzeni, « 3D digitizing of cultural heritage », *Journal of Cultural Heritage*, vol. 2, *1*, pp. 63–70, 2001.

[4] C. Wang, Y. K. Cho, and C. Kim, « Automatic BIM component extraction from point clouds of existing buildings for sustainability applications », *Automation in Construction*, vol. 56, pp. 1–13, 2015.

[5] N. Hichri, C. Stefani, L. De Luca, P. Veron, and G. Hamon, « From point cloud to BIM: a survey of existing approaches », *in XXIV International CIPA Symposium*, strasbourg, France: Proceedings of the XXIV International CIPA Symposium, 2013.

[6] P. Rosenthal and L. Linsen, « Image-space point cloud rendering », *in Proceedings of Computer Graphics International*, 2008, pp. 136–143.

[7] I. Goudé, R. Cozot, and O. Le Meur, « A Perceptually Coherent TMO for Visualization of 360° HDR Images on HMD », *Transactions on Computational Science XXXVII*, vol. 37, pp. 109–128, 2020.

[8] I. Goudé, R. Cozot, and F. Banterle, « HMD-TMO: A Tone Mapping Operator for 360° HDR Images Visualization for Head Mounted Displays », *in Computer Graphics International Conference*, Springer, 2019, pp. 216–227.

[9] I. Goudé, J. Lacoche, and R. Cozot, « Tone mapping high dynamic 3D scenes with global lightness coherency », *Computers & Graphics*, vol. 91, pp. 243–251, 2020.

[10] M. D. Fairchild, *Color appearance models*. John Wiley & Sons, 2013.

[11] D. C. Hood and M. A. Finkelstein, « Sensitivity to light », *Handbook of Perception and Human Performance (Vol. 1: Sensory Processes and Perception)*, vol. 1, 1986.

[12] T. Young, « II. The Bakerian Lecture. On the theory of light and colours », *Philosophical transactions of the Royal Society of London*, *92*, pp. 12–48, 1802.

[13] W. D. Wright, « A re-determination of the trichromatic coefficients of the spectral colours », *Transactions of the Optical Society*, vol. 30, *4*, p. 141, 1929.

[14] J. Guild, « The colorimetric properties of the spectrum », *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 230, *681-693*, pp. 149–187, 1931.

[15] *Recommandation ITU-R BT.2020-2, Parameter values for ultra-high definition television systems for productionand international programme exchange*, Oct. 2015.

[16] G. H. Joblove and D. Greenberg, « Color spaces for computer graphics », *in Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, 1978, pp. 20–25.

[17] *ISO/CIE 11664-4:2019, Colorimétrie — Partie 4: Espace chromatique L*a*b* CIE 1976*, Jun. 2019.

[18] J. A. Ferwerda, S. N. Pattanaik, P. Shirley, and D. P. Greenberg, « A model of visual adaptation for realistic image synthesis », *in Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM, 1996, pp. 249–258.

[19] E. Reinhard, W. Heidrich, P. Debevec, S. Pattanaik, G. Ward, and K. Myszkowski, *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010.

[20] F. Banterle, A. Artusi, K. Debattista, and A. Chalmers, *Advanced high dynamic range imaging*. AK Peters/CRC Press, 2017.

[21] G. Ward, « Real pixels », *Graphics Gems II*, vol. 2, pp. 80–83, 1991.

[22] G. Ward and M. Simmons, « Subband encoding of high dynamic range imagery », *in Proceedings of the 1st Symposium on Applied Perception in Graphics and Visualization*, 2004, pp. 83–90.

[23] R. Mantiuk, G. Krawczyk, K. Myszkowski, and H.-P. Seidel, « Perception-motivated high dynamic range video encoding », *ACM Transactions on Graphics (TOG)*, vol. 23, *3*, pp. 733–741, 2004.

[24] R. Xu, S. N. Pattanaik, and C. E. Hughes, « High-dynamic-range still-image encoding in JPEG 2000 », *IEEE Computer Graphics and Applications*, vol. 25, *6*, pp. 57–64, 2005.

[25] Z. Mai, H. Mansour, R. Mantiuk, P. Nasiopoulos, R. Ward, and W. Heidrich, « Optimizing a tone curve for backward-compatible high dynamic range image and video compression », *IEEE transactions on image processing*, vol. 20, *6*, pp. 1558–1571, 2010.

[26] R. Mantiuk, K. J. Kim, A. G. Rempel, and W. Heidrich, « HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions », *ACM Transactions on graphics (TOG)*, vol. 30, *4*, pp. 1–14, 2011.

[27] M. Rousselot, E. Auffret, X. Ducloux, O. Le Meur, and R. Cozot, « Impacts of viewing conditions on hdr-vdp2 », *in 2018 26th European Signal Processing Conference (EUSIPCO)*, IEEE, 2018, pp. 1442–1446.

[28] M. Rousselot, X. Ducloux, O. Le Meur, and R. Cozot, « Quality metric aggregation for HDR/WCG images », *in 2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, pp. 3786–3790.

[29] M. Rousselot, O. Le Meur, R. Cozot, and X. Ducloux, « Quality assessment of hdr/wcg images using hdr uniform color spaces », *Journal of Imaging*, vol. 5, *1*, p. 18, 2019.

[30] F. Banterle, A. Artusi, A. Moreo, and F. Carrara, « Nor-Vdpnet: A No-Reference High Dynamic Range Quality Metric Trained On Hdr-Vdp 2 », *in 2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2020, pp. 126–130.

[31] C. Schlick, « Quantization techniques for visualization of high dynamic range pictures », *in Photorealistic Rendering Techniques*, Springer, 1995, pp. 7–20.

[32] J. Tumblin, J. K. Hodgins, and B. K. Guenter, « Two methods for display of high contrast images », *ACM Transactions on Graphics (TOG)*, vol. 18, *1*, pp. 56–94, 1999.

[33] G. W. Larson, H. Rushmeier, and C. Piatko, « A visibility matching tone reproduction operator for high dynamic range scenes », *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, *4*, pp. 291–306, 1997.

[34] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, « Adaptive logarithmic mapping for displaying high contrast scenes », *in Computer graphics forum*, Wiley Online Library, vol. 22, 2003, pp. 419–426.

[35] S. N. Pattanaik, J. A. Ferwerda, M. D. Fairchild, and D. P. Greenberg, « A multiscale model of adaptation and spatial vision for realistic image display », *in Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998, pp. 287–298.

[36] M. Ashikhmin, « A Tone Mapping Algorithm for High Contrast Images », *in Proceedings of the 13th Eurographics Workshop on Rendering*, ser. EGRW '02, Pisa, Italy: Eurographics Association, 2002, pp. 145–156.

[37] R. Mantiuk, K. Myszkowski, and H.-P. Seidel, « A perceptual framework for contrast processing of high dynamic range images », *ACM Transactions on Applied Perception (TAP)*, vol. 3, *3*, pp. 286–308, 2006.

[38] S. N. Pattanaik, J. Tumblin, H. Yee, and D. P. Greenberg, « Time-dependent visual adaptation for fast realistic image display », *in Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 2000, pp. 47–54.

[39] P. Ledda, L. P. Santos, and A. Chalmers, « A local model of eye adaptation for high dynamic range images », *in Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, ACM, 2004, pp. 151–160.

[40] F. Durand and J. Dorsey, « Fast bilateral filtering for the display of high-dynamic-range images », *in Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 2002, pp. 257–266.

[41] J. Kuang, G. M. Johnson, and M. D. Fairchild, « iCAM06: A refined image appearance model for HDR image rendering », *Journal of Visual Communication and Image Representation*, vol. 18, *5*, pp. 406–414, 2007.

[42] G. Krawczyk, K. Myszkowski, and H.-P. Seidel, « Lightness perception in tone reproduction for high dynamic range images », *in Computer Graphics Forum*, Citeseer, vol. 24, 2005, pp. 635–646.

[43] R. Boitard, R. Cozot, D. Thoreau, and K. Bouatouch, « Survey of temporal brightness artifacts in video tone mapping », *in HDRi2014-Second International Conference and SME Workshop on HDR imaging*, vol. 9, 2014.

[44] ——, « Zonal brightness coherency for video tone mapping », *Signal Processing: Image Communication*, vol. 29, *2*, pp. 229–246, 2014.

[45] T. O. Aydin, N. Stefanoski, S. Croci, M. Gross, and A. Smolic, « Temporally coherent local tone mapping of HDR video », *ACM Transactions on Graphics (TOG)*, vol. 33, *6*, pp. 1–13, 2014.

[46] G. Eilertsen, R. K. Mantiuk, and J. Unger, « Real-time noise-aware tone mapping », *ACM Transactions on Graphics (TOG)*, vol. 34, p. 198, 2015.

[47] ——, « A comparative review of tone-mapping algorithms for high dynamic range video », *in Computer Graphics Forum*, Wiley Online Library, vol. 36, 2017, pp. 565–592.

[48] M. Yu, « Dynamic tone mapping with head-mounted displays », *Standford University Report*, vol. 5, 2015.

[49] M. Gross and H. Pfister, *Point-based graphics*. Elsevier, 2011.

[50] R. A. Hamzah and H. Ibrahim, « Literature survey on stereo vision disparity map algorithms », *Journal of Sensors*, vol. 2016, 2016.

[51] D. Eigen and R. Fergus, « Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture », *in Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650–2658.

[52] F. Liu, C. Shen, G. Lin, and I. Reid, « Learning depth from single monocular images using deep convolutional neural fields », *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, *10*, pp. 2024–2039, 2015.

[53] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, « Deeper depth prediction with fully convolutional residual networks », *in 2016 Fourth international conference on 3D vision (3DV)*, IEEE, 2016, pp. 239–248.

[54] X. Luo, J.-B. Huang, R. Szeliski, K. Matzen, and J. Kopf, « Consistent video depth estimation », *arXiv preprint arXiv:2004.15021*, 2020.

[55] G. Turk, « The PLY polygon file format », 1994.

[56] R. B. Rusu and S. Cousins, « 3d is here: Point cloud library (pcl) », *in 2011 IEEE international conference on robotics and automation*, IEEE, 2011, pp. 1–4.

[57] C. Zhang, D. Florencio, and C. Loop, « Point cloud attribute compression with graph transform », *in 2014 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2014, pp. 2066–2070.

[58] D. Thanou, P. A. Chou, and P. Frossard, « Graph-based compression of dynamic 3D point cloud sequences », *IEEE Transactions on Image Processing*, vol. 25, *4*, pp. 1765–1778, 2016.

[59] M. Quach, G. Valenzise, and F. Dufaux, « Learning convolutional transforms for lossy point cloud geometry compression », *in 2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, pp. 4320–4324.

[60] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, « Geometric distortion metrics for point cloud compression », *in 2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2017, pp. 3460–3464.

[61] E. Alexiou and T. Ebrahimi, « On subjective and objective quality evaluation of point cloud geometry », *in 2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, IEEE, 2017, pp. 1–3.

[62] ——, « Point cloud quality assessment metric based on angular similarity », *in 2018 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2018, pp. 1–6.

[63] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li, *et al.*, « Emerging MPEG standards for point cloud compression », *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, *1*, pp. 133–148, 2018.

[64] S. Perry, H. P. Cong, L. A. da Silva Cruz, J. Prazeres, M. Pereira, A. Pinheiro, E. Dumic, E. Alexiou, and T. Ebrahimi, « Quality evaluation of static point clouds encoded using MPEG codecs », *in 2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2020, pp. 3428–3432.

[65] H. Pfister, M. Zwicker, J. Van Baar, and M. Gross, « Surfels: Surface elements as rendering primitives », *in Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 335–342.

[66] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, « Surface splatting », *in Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 371–378.

[67] M. Schütz and M. Wimmer, « High-quality point-based rendering using fast single-pass interpolation », *in 2015 Digital Heritage*, IEEE, vol. 1, 2015, pp. 369–372.

[68] R. K. dos Anjos, C. S. Ribeiro, D. S. Lopes, and J. M. Pereira, « Stroke-based splatting: an efficient multi-resolution point cloud visualization technique », *The Visual Computer*, vol. 34, *10*, pp. 1383–1397, 2018.

[69] Y. Quinsat and C. Lartigue, « Filling holes in digitized point cloud using a morphing-based approach to preserve volume characteristics », *The International Journal of Advanced Manufacturing Technology*, vol. 81, *1-4*, pp. 411–421, 2015.

[70] Y. Jun, « A piecewise hole filling algorithm in reverse engineering », *Computer-aided design*, vol. 37, *2*, pp. 263–270, 2005.

[71] C. Moenning and N. A. Dodgson, « Intrinsic point cloud simplification », *Proc. 14th GrahiCon*, vol. 14, p. 23, 2004.

[72] S. Rusinkiewicz and M. Levoy, « QSplat: A multiresolution point rendering system for large meshes », *in Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 343–352.

[73] ——, « Streaming QSplat: A viewer for networked visualization of large, dense models », *in Proceedings of the 2001 symposium on Interactive 3D graphics*, 2001, pp. 63–68.

[74] M. Schütz, « Potree: Rendering large point clouds in web browsers », *Technische Universität Wien, Wiedeń*, 2016.

[75] M. Levoy and T. Whitted, *The use of points as a display primitive*. Citeseer, 1985.

[76] G. T. Fechner, D. H. Howes, and E. G. Boring, *Elements of psychophysics*. Holt, Rinehart and Winston New York, 1966, vol. 1.

[77] S. S. Stevens, « To honor Fechner and repeal his law », *Science*, vol. 133, *3446*, pp. 80–86, 1961.

[78] M. H. Brill and R. C. Carter, « Does lightness obey a log or a power law? Or is that the right question? », *Color Research & Application*, vol. 39, *1*, pp. 99–101, 2014.

[79] P. Whittle, « Increments and decrements: luminance discrimination », *Vision research*, vol. 26, *10*, pp. 1677–1691, 1986.

[80] C. Li, Z. Li, Z. Wang, Y. Xu, M. R. Luo, G. Cui, M. Melgosa, M. H. Brill, and M. Pointer, « Comprehensive color solutions: CAM16, CAT16, and CAM16-UCS », *Color Research & Application*, vol. 42, *6*, pp. 703–718, 2017.

[81] A. H. Gloriani, B. M. Matesanz, P. A. Barrionuevo, I. Arranz, L. Issolio, S. Mar, and J. A. Aparicio, « Influence of background size, luminance and eccentricity on different adaptation mechanisms », *Vision research*, vol. 125, pp. 12–22, 2016.

[82] S.-H. Lee and K.-I. Sohng, « A Model of Luminance-Adaptation for Quantifying Brightness in Mixed Visual Adapting Conditions », *IEICE transactions on electronics*, vol. 94, *11*, pp. 1768–1772, 2011.

[83] A.-F. Perrin, C. Bist, R. Cozot, and T. Ebrahimi, « Measuring quality of omnidirectional high dynamic range content », *in Applications of Digital Image Processing XL*, International Society for Optics and Photonics, vol. 10396, 2017, p. 1 039 613.

[84] M. Melo, K. Bouatouch, M. Bessa, H. Coelho, R. Cozot, and A. Chalmers, « Tone Mapping HDR Panoramas for Viewing in Head Mounted Displays. », *in VISIGRAPP (1: GRAPP)*, 2018, pp. 232–239.

[85] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, « Photographic tone reproduction for digital images », *in ACM transactions on graphics (TOG)*, ACM, vol. 21, 2002, pp. 267–276.

[86] S. Cutchin and Y. Li, « View dependent tone mapping of HDR panoramas for head mounted displays », *in Proceedings of the 26th International Conference on Artificial Reality and Telexistence and the 21st Eurographics Symposium on Virtual Environments*, Eurographics Association, 2016, pp. 29–36.

[87] *Recommandation ITU-R BT.709-6, Parameter values for the HDTV standards for production and international programme exchange*, Jun. 2015.

[88] E. Upenik and T. Ebrahimi, « A simple method to obtain visual attention data in head mounted virtual reality », *in 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE, 2017, pp. 73–78.

[89] A. De Abreu, C. Ozcinar, and A. Smolic, « Look around you: Saliency maps for omnidirectional images in VR applications », *in 2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, IEEE, 2017, pp. 1–6.

[90] R. Mantiuk, R. Mantiuk, A. Tomaszewska, and W. Heidrich, « Color correction for tone mapping », *in Computer Graphics Forum*, Wiley Online Library, vol. 28, 2009, pp. 193–202.

[91] A. Artusi, T. Pouli, F. Banterle, and A. O. Akyüz, « Automatic saturation correction for dynamic range management algorithms », *Signal Processing: Image Communication*, vol. 63, pp. 100–112, 2018.

[92] H. Yeganeh and Z. Wang, « Objective quality assessment of tone-mapped images », *IEEE Transactions on Image Processing*, vol. 22, *2*, pp. 657–667, 2012.

[93] D. Pajak, M. Cadìk, T. O. Aydin, K. Myszkowski, and H.-P. Seidel, « Visual maladaptation in contrast domain », *in Human Vision and Electronic Imaging XV*, International Society for Optics and Photonics, vol. 7527, 2010, p. 752 710.

[94] T. Ritschel and E. Eisemann, « A computational model of afterimages », *in Computer Graphics Forum*, Wiley Online Library, vol. 31, 2012, pp. 529–534.

[95] D. E. Jacobs, O. Gallo, E. A. Cooper, K. Pulli, and M. Levoy, « Simulating the visual experience of very bright and very dark scenes », *ACM Transactions on Graphics (TOG)*, vol. 34, *3*, pp. 1–15, 2015.

[96] F. Durand and J. Dorsey, « Interactive tone mapping », *in Rendering Techniques 2000*, Springer, 2000, pp. 219–230.

[97] E. H. Adelson, J. R. Bergen, *et al.*, *The plenoptic function and the elements of early vision*. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1991, vol. 2.

[98] L. McMillan and G. Bishop, « Plenoptic modeling: An image-based rendering system », *in Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 39–46.

[99] P. Debevec, « Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography », *in ACM SIGGRAPH 2008 classes*, ACM, 2008, p. 32.

[100] S. LaValle, « Virtual reality », 2016.

[101] J. J. LaViola Jr, « A discussion of cybersickness in virtual environments », *ACM Sigchi Bulletin*, vol. 32, *1*, pp. 47–56, 2000.

[102] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal, « Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness », *The international journal of aviation psychology*, vol. 3, *3*, pp. 203–220, 1993.

[103] X. Cao, W. Wang, K. Nagao, and R. Nakamura, « PSNet: A Style Transfer Network for Point Cloud Stylization on Geometry and Color », *in The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 3337–3345.

[104] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, « Pointnet: Deep learning on point sets for 3d classification and segmentation », *in Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[105] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, « Color transfer between images », *IEEE Computer graphics and applications*, vol. 21, *5*, pp. 34–41, 2001.

[106] D. L. Ruderman, T. W. Cronin, and C.-C. Chiao, « Statistics of cone responses to natural images: implications for visual coding », *JOSA A*, vol. 15, *8*, pp. 2036–2045, 1998.

[107] F. Pitié and A. Kokaram, « The linear monge-kantorovitch linear colour mapping for example-based colour transfer », 2007.

[108] H. Hristova, O. Le Meur, R. Cozot, and K. Bouatouch, « Style-aware robust color transfer. », *in Computational Aesthetics*, 2015, pp. 67–77.

[109] H. Hristova, O. Le Meur, R. Cozot, and K. Bouatouch, « Transformation of the multivariate generalized Gaussian distribution for image editing », *IEEE transactions on visualization and computer graphics*, vol. 24, *10*, pp. 2813–2826, 2017.

[110] H. Hristova, O. Le Meur, R. Cozot, and K. Bouatouch, « Transformation of the beta distribution for color transfer », *in VISIGRAPP (GRAPP)*, 2018.

[111] L. A. Gatys, A. S. Ecker, and M. Bethge, « Image style transfer using convolutional neural networks », *in Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[112] J. Johnson, A. Alahi, and L. Fei-Fei, « Perceptual losses for real-time style transfer and super-resolution », *in European conference on computer vision*, Springer, 2016, pp. 694–711.

[113] B. Kim, V. C. Azevedo, M. Gross, and B. Solenthaler, « Lagrangian Neural Style Transfer for Fluids », *ACM Transactions on Graphics*, vol. 39, *4*, 2020. DOI: 10.1145/3386569.3392473.

[114] O. Texler, D. Futschik, M. Kučera, O. Jamriška, Š. Sochorová, M. Chai, S. Tulyakov, and D. Sykora, « Interactive Video Stylization Using Few-Shot Patch-Based Training », *arXiv preprint arXiv:2004.14489*, 2020.

[115] J. P. Grossman and W. J. Dally, « Point sample rendering », *in Rendering techniques' 98*, Springer, 1998, pp. 181–192.

[116] H. Xu, M. X. Nguyen, X. Yuan, and B. Chen, « Interactive silhouette rendering for point-based models », *in Proceedings of the First Eurographics conference on Point-Based Graphics*, Eurographics Association, 2004, pp. 13–18.

[117] T. Mertens, J. Kautz, J. Chen, P. Bekaert, and F. Durand, « Texture Transfer Using Geometry Correlation. », *Rendering Techniques*, vol. 273, *10.2312*, pp. 273–284, 2006.

[118] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, « Deep learning for 3d point clouds: A survey », *arXiv preprint arXiv:1912.12033*, 2019.

[119] X. Cao and K. Nagao, « Point cloud colorization based on densely annotated 3D shape dataset », *in International Conference on Multimedia Modeling*, Springer, 2019, pp. 436–446.

[120] N. Bonneel, G. Peyré, and M. Cuturi, « Wasserstein barycentric coordinates: histogram regression using optimal transport. », *ACM Trans. Graph.*, vol. 35, *4*, pp. 71–1, 2016.

[121] Y. Mroueh, « Wasserstein style transfer », *arXiv preprint arXiv:1905.12828*, 2019.

[122] E. Alexiou and T. Ebrahimi, « Exploiting user interactivity in quality assessment of point cloud imaging », *in 2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, IEEE, 2019, pp. 1–6.

[123] S. Mérillou and D. Ghazanfarpour, « A survey of aging and weathering phenomena in computer graphics », *Computers & Graphics*, vol. 32, *2*, pp. 159–174, 2008.

UNIVERSITÉ DE RENNES 1

Titre : Rendu de nuages de points 3D à Grande Gamme Dynamique

Mot clés : Nuage de point, Grande Gamme Dynamique, Opérateur de mappage de ton, Casque de Réalité Virtuelle, Transfert de couleur

Résumé : Cette thèse s'intéresse à l'amélioration de la qualité de rendu de nouveaux formats 3D et HDR (à grande gamme dynamique). L'aspect tridimensionnel (géométrie de la scène) allié au HDR permet une représentation plus fidèle de la luminance et de la couleur des objets d'une scène 3D. L'objectif principal est de proposer des solutions d'affichage de nuages de points 3D HDR sur des écrans ayant des caractéristiques variées. Cet objectif est double. Le premier objectif concerne le rendu de contenu 3D HDR sur des écrans classiques. Les solutions que nous proposons permettent d'améliorer la qualité du rendu de contenus (nuages de points 3D HDR) sur des écrans classiques et sur des casques de réalité virtuelle. Ces améliorations résultent des évaluations subjectives que nous avons menées sur la perception des couleurs sur des casques de réalité virtuelle. Le deuxième objectif concerne la stylisation de contenus 3D représentés par un nuage de points. Alors qu'il existe beaucoup de techniques pour styliser des images (filtres, effets de flou ou de vignette, etc.), la stylisation de contenus 3D a été très peu étudiée. Nous proposons une méthode de stylisation consistant à transférer la couleur d'un nuage de points à un autre. Cette méthode est basée sur l'exemple et prend en compte la géométrie des nuages de points. Nos résultats et évaluations ont montré une amélioration significative par rapport aux méthodes de transfert de couleur existantes.

Title: Rendering of High Dynamic Range 3D point clouds

Keywords: Point cloud, High Dynamic Range, Tone Mapping Operator, Head Mounted Display, Color transfer

Abstract: This thesis addresses the improvement of the quality of rendering new 3D and HDR (High Dynamic Range) format. The 3D aspect (geometry of the scene) together with HDR allows a more faithful representation of the luminance and the color of the objects within a 3D scene. The main objective is to propose solutions to display HDR 3D point clouds on display units of various characteristics. The objective is twofold. The first objective is concerned with the rendering of HDR 3D contents on mainstream displays. The solutions we propose allow improving the quality of the rendering of contents (HDR 3D point clouds) on mainstream displays and HMDs (Head Mounted Displays). This improvement result from subjective evaluations we have conducted on the perception of color on HMDs. The second objective is the stylization of 3D contents represented by point clouds. While there exist many stylization techniques applied to images (filters, blurring or vignetting effects, etc.), the stylization of 3D contents has aroused little interest. For this reason, we propose a stylization method consisting of transferring the color of a point cloud to another. This method is example-based and accounts for the geometry of the point clouds. Our results and evaluations have shown a significant improvement compared to existing color transfer methods.