

UNIVERSITÄT
DES
SAARLANDES

Design and Applications of Perception-Based Mesh, Image, and Display-Related Quality Metrics

Dissertation zur Erlangung des Grades des Doktors der Ingenieurwissenschaften der Fakultät für Mathematik und Informatik der Universität des Saarlandes

Vorgelegt von
Krzysztof Wolski

Saarbrücken, 2023

Dean:	Prof. Dr. Jürgen Steimle
Date:	December 4th, 2023
Chair:	Prof. Dr.-Ing. Thorsten Herfet
Reviewers:	Prof. Dr. Karol Myszkowski
	Prof. Dr. Hans-Peter Seidel
	Prof. Dr. Carol O'Sullivan
	Prof. Dr. Rafał Mantiuk
Academic Assistant:	Dr. Thomas Leimkühler

Abstract

Computer graphics have become an integral part of our daily lives, enabling immersive experiences in movies, video games, virtual reality, and augmented reality. However, the various stages of the computer graphics pipeline, from content generation to rendering and display, present their own challenges that can reduce visual quality and thus degrade the overall experience. Perceptual metrics are crucial for evaluating visual quality. However, many existing methods have limitations in reproducing human perception accurately, as they must account for the complexities of the human visual system. This dissertation aims to tackle these issues by proposing innovative advancements across different pipeline stages. Firstly, it introduces a novel neural-based visibility metric to improve the assessment of near-threshold image distortions. Secondly, it addresses shortcomings of the mesh quality metrics, vital for enhancing the integrity of three-dimensional models. Thirdly, the dissertation focuses on optimizing the visual quality of animated content while considering display characteristics and a limited rendering budget. Finally, the work delves into the challenges specific to stereo vision in a virtual reality setting. The ultimate objective is to enable the creation of more efficient and automated designs for virtual experiences, benefiting fields like entertainment and education. Through these contributions, this research seeks to elevate the standard of visual quality in computer graphics, enriching the way we interact with virtual worlds.

Zusammenfassung

Computergrafik ist zu einem integralen Bestandteil unseres täglichen Lebens geworden und ermöglicht immersive Erlebnisse in Filmen, Videospielen, virtueller und erweiterter Realität. Allerdings stellen die verschiedenen Phasen der Computergrafik-Pipeline, angefangen vom Erstellen von Inhalten bis hin zu deren Rendering und Anzeige, ihre eigenen Herausforderungen dar, die die visuelle Qualität beeinträchtigen und somit das Gesamterlebnis negative beeinträchtigen können. Perzeptionsmetriken sind entscheidend für die Bewertung der visuellen Qualität. Viele bestehende Methoden haben jedoch ihre Grenzen bei der genauen Wiedergabe der menschlichen Wahrnehmung, da sie die Komplexität des menschlichen visuellen Systems berücksichtigen müssen. Diese Dissertation zielt darauf ab, diese Probleme zu lösen, indem sie innovative Fortschritte in verschiedenen Phasen der Pipeline vorschlägt. Zunächst wird eine neuartige, neuronale Sichtbarkeitsmetrik vorgestellt, um die Beurteilung von Bildverzerrungen nahe der Wahrnehmungsschwelle zu verbessern. Zweitens werden Schwächen von Mesh-Qualitätsmetriken behoben, die für verzerrungsfreie dreidimensionale Modelle von entscheidender Bedeutung sind. Drittens konzentriert sich die Dissertation darauf, die visuelle Qualität von animierten Inhalten zu optimieren und gleichzeitig Bildschirmeigenschaften und ein begrenztes Rendering-Budget zu berücksichtigen. Schließlich widmet sich die Arbeit den spezifischen Herausforderungen der Stereo-Vision in der virtuellen Realität. Das ultimative Ziel ist es, die Schaffung effizienterer und automatisierter Designs für virtuelle Erfahrungen zu ermöglichen, was Bereiche wie Unterhaltung und Bildung zugutekommt. Durch diese Beiträge strebt diese Forschung an, den Standard der visuellen Qualität in der Computergrafik zu erhöhen und unsere Interaktionen mit virtuellen Welten zu bereichern.

Acknowledgments

I am filled with immense gratitude to all the individuals I met on my doctoral journey, as they have been the cornerstone of this work. First and foremost, I express my deepest appreciation to Karol Myszkowski, who has been the guiding beacon throughout my academic pursuits. His expertise, patience, and constructive feedback have been the driving force behind my growth and accomplishments. I also want to thank my co-advisors, Rafał Mantiuk and Piotr Didyk, whose inestimable contributions have offered unique insights and perspectives that enriched my understanding of the subject matter. I express my appreciation to Hans-Peter Seidel for providing me with remarkable opportunities to develop my scientific skills and expertise.

The period of my Ph.D. would have undeniably lacked something important without the presence of my exceptional peers and colleagues, who have consistently stood by me to share my triumphs and challenges. I wish to extend my heartfelt gratitude to Mojtaba Bemana, Thomas Leimkühler, Rhaleb Zayer, and Vamsi Kiran Adhikarla for their unwavering camaraderie and support. I am also deeply indebted to all members of AG4 for enriching my academic and personal life.

Without a doubt, all my achievements would not have been possible without the exceptional collaborations I have had the opportunity to engage in. Hence, I want to thank Akshay Jindal, Daniele Giunchi, and Fangcheng Zhong for their contributions and all the time spent together.

Moreover, my appreciation would be remiss if I did not mention Alex Chapiro, to whom I owe a debt of gratitude for his invaluable support and guidance during my internship at Reality Labs. During this internship, I had the privilege of engaging in fruitful interactions with esteemed colleagues at Meta. Among them, Minjung Kim and Trisha Lian hold a special place, and I wish to express my gratitude to them, not solely for their scientific assistance but also for our inspiring and joyful art meetings that continue to this day.

Furthermore, I thank Piotr Brzozowski, a cherished friend from my school days, for his continuous support.

Last but certainly not least, I extend my profound thanks to my parents and family for their constant belief in me and their assistance throughout every stage of my life.

Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Contributions . . . 4
1.3 Outline . . . 6

2 Background and Previous Work **7**
2.1 Human Visual System . . . 7
2.1.1 Anatomy of Human Visual System . . . 7
2.1.2 Contrast Perception . . . 10
2.1.3 Motion Perception . . . 13
2.1.4 Depth Perception . . . 16
2.1.5 Perception in Low Luminance Conditions . . . 20
2.1.6 Masking . . . 22
2.2 State-of-the-Art Metrics . . . 23
2.2.1 Image Quality and Visibility Metrics . . . 23
2.2.2 Mesh Quality Metrics . . . 27
2.2.3 Motion Quality Metrics . . . 28
2.3 Technical Aspects and Tools . . . 29
2.3.1 Perceptual Scaling . . . 29
2.3.2 Variable Rate Shading . . . 29

3 General Purpose Metric Design **31**
3.1 Introduction . . . 31
3.2 Dataset of Visible Distortions . . . 32
3.3 Experimental Procedure and Apparatus . . . 37
3.4 Modeling Experimental Data . . . 39
3.5 Visibility Metrics . . . 42
3.6 CNN-Based Metric . . . 44
3.6.1 Two-Branch Fully Convoluted Architecture . . . 44
3.6.2 Training and Testing . . . 45
3.7 Metric Results . . . 46
3.8 Applications . . . 48
3.8.1 Visually Lossless Image Compression . . . 48
3.8.2 Super-Resolution from Downsampled Images . . . 49
3.8.3 Content-Adaptive Watermarking . . . 50
3.9 Extension for Viewing Distance and Display Brightness . . . 51
3.9.1 Data Collection . . . 52
3.9.2 Metric Architecture . . . 53
3.9.3 Training . . . 55
3.9.4 Results . . . 56
3.10 Limitations . . . 57
3.11 Conclusions . . . 57

4 Selecting Texture Resolution Using a Task-Specific Visibility Metric 61
4.1 Introduction . . . 61
4.2 Overview . . . 62
4.3 Background . . . 63
4.3.1 Level of Detail for Textured 3D Models . . . 63
4.3.2 Texture Compression Masking . . . 64
4.4 Experiment 1: Detection Thresholds . . . 65
4.4.1 Cross-Population Psychometric Function . . . 67
4.5 Experiment 2: Local Visibility Maps . . . 68
4.6 Pooling for Task-Specific Visibility Threshold . . . 69
4.7 Metric Retuning . . . 71
4.7.1 Retuning Dataset . . . 71
4.7.2 Loss Function . . . 73
4.7.3 Transfer Learning and Testing Performance . . . 73
4.8 Results . . . 73
4.8.1 Visibility Map Improvement . . . 74
4.9 Texture Size Optimization . . . 75
4.10 Limitations . . . 78
4.11 Conclusions . . . 79

5 Perceptual Dataset of Mesh Distortions 81
5.1 Introduction . . . 81
5.2 Face Perception and Datasets . . . 82
5.3 Synthetic Dataset . . . 83
5.3.1 Non-Public Dataset . . . 83
5.3.2 Public Synthetic Heads Generation . . . 83
5.3.3 Distortion Dataset . . . 84
5.4 User Study . . . 88
5.4.1 Participants . . . 88
5.4.2 Stimulus Rendering . . . 89
5.4.3 Experimental Procedure . . . 89
5.4.4 Data Analysis . . . 91
5.5 Experiment Results . . . 91
5.6 Distortion Metric Analysis . . . 93
5.7 Applications . . . 93
5.7.1 Retargeting Metrics . . . 94
5.7.2 Perceptually Bound Compression . . . 95
5.7.3 Perceptually Driven Automatic LOD . . . 96
5.8 Conclusions . . . 98

6 Perceptual Model for Adaptive Local Shading and Refresh Rate 99
6.1 Introduction . . . 99
6.2 Background . . . 100
6.2.1 Motion Adaptive Rendering . . . 100
6.2.2 Fixed-Bandwidth Rendering . . . 100
6.3 CaMoJAB: Content-adaptive Metric of Judder, Aliasing and Blur . . . 101
6.3.1 Spatial Distortion Model . . . 101
6.3.2 Temporal Distortion Model . . . 107
6.3.3 Experiment 1: VRS Motion Quality Measurement . . . 107
6.3.4 Model Calibration . . . 110
6.3.5 CaMoJAB Results and Comparison . . . 110

6.3.6 CaMoJAB Validation . . . 112
6.4 Adaptive Local Shading and Refresh Rate . . . 113
6.4.1 Auxiliary Buffers . . . 114
6.4.2 Optimal Refresh Rate . . . 114
6.4.3 Optimal Shading Rates . . . 115
6.4.4 The Velocity of Eye Motion Without Eye Tracking . . . 117
6.4.5 Implementation Details . . . 117
6.4.6 Experiment 2: ALSaRR for Real-Time Rendering . . . 117
6.5 Limitations and Discussion . . . 120
6.6 Conclusions . . . 121

7 Improving Depth Perception Under Low Luminance 123
7.1 Introduction . . . 123
7.2 Experiment 1: 3D Shape Perception . . . 124
7.3 Stereo Constancy Model . . . 127
7.4 Stereo-Preserving Contrast Enhancement Method . . . 128
7.4.1 Color Space Transformation . . . 129
7.4.2 Multi-Scale Decomposition . . . 130
7.4.3 Measure of Local Contrast . . . 130
7.4.4 Contrast Retargeting . . . 131
7.4.5 Reconstructing Color Image . . . 133
7.4.6 Implementation Details . . . 133
7.5 Experiment 2: Validation . . . 134
7.6 Discussion . . . 137
7.7 Conclusions . . . 138

8 Conclusion 139
8.1 Discussion . . . 139
8.2 Future Work . . . 142
8.3 Closing Remarks . . . 146

Bibliography - Own Work 147

Bibliography 149

Chapter 1

Introduction

1.1 Motivation



FIGURE 1.1: *Distortions that could potentially degrade the overall quality of the final output can arise at various stages of the computer graphics pipeline. During the asset creation phase, inadequate optimization of mesh topology can result in mesh artifacts (left). Subsequently, the rendered image may undergo a visually-excessive compression for storage or streaming purposes (middle). Finally, when displaying moving content, numerous motion artifacts, such as hold-type blur, can arise (right).*

Computer graphics have become an integral part of our daily lives, enriching the domain of cinematography and fostering a plethora of diverse experiences, ranging from video games to applications of Virtual Reality (VR) and Augmented Reality (AR). Among these, real-time computer graphics have had a particularly profound impact, enabling users to engage with virtual environments. As a result, these applications have opened up new horizons in fields like education, healthcare, and entertainment, delivering immersive and captivating experiences.

The traditional computer graphics pipeline comprises multiple stages that transform abstract artistic concepts into compelling visual stimuli. It all starts with content generation, where talented artists create digital assets like 3D models, textures, and animations. Once these assets are prepared, the rendering stage begins, employing intricate algorithms to simulate physics-based light transport, ultimately transforming analytical scene descriptions into final two-dimensional images or animations, which are then stored in buffers or as digital files. In the final stage, the rendered content is displayed to end users, utilizing a range of devices such as computer monitors, television sets, virtual reality headsets, and mobile devices.

Despite the great strides, each of these pipeline stages carries the risk of introducing artifacts that can adversely affect the visual quality (Figure 1.1). For example, on the geometry level, the quality can be influenced by the used data sources, where 3D scans and meshes generated through photogrammetry or automated methods may contain noise or errors. Moreover, contamination may arise from processes such as mesh compression. While this method is commonly employed to reduce data size and improve rendering performance, it can result in information loss and, consequently, artifacts on the mesh surface. Additionally, commonly used geometry

processing algorithms like mesh simplification (which serves as a basis for level-of-detail technique), smoothing, or subdivision can also produce undesirable side effects if not applied with care. As a result, considerable post-processing is often required to rectify these inaccuracies or distortions.

Moving to the rendering stage, striking a balance between performance and image quality becomes critical due to the computationally intensive and time-consuming nature of rendering. To enhance performance, various shortcuts and optimizations may be implemented. For instance, in offline rendering, reducing the number of samples in path tracing can significantly shorten rendering time but may result in noisy output. In real-time rendering, where global illumination models often crudely approximate real-life phenomena, there are additional challenges in maintaining visual quality. A prominent illustration of this is shadow mapping, a widely-employed technique for real-time rendering of shadows. While it effectively reduces computational resource usage, it comes with the drawback of producing artifacts such as low-resolution shadows and light leaks, ultimately compromising the visual quality of the rendered scene.

At the final stage, when visual content is presented to the end user, it becomes crucial to account for the parameters of the displaying hardware. This is particularly important when presenting animated content, as its motion can effectively conceal spatial distortions in the visual output, but it may also introduce new distortions in the temporal domain. To ensure a high-quality user experience, careful attention must be given to the display properties of the output device, including refresh rate, display persistence, resolution, brightness, and display type. Achieving a flawless visual experience in a VR headset can pose even greater challenges compared to a standard display, as an inadequate selection of the mentioned parameters may result in the loss of cues for stereopsis or even cause motion sickness.

It is undeniable that understanding how algorithmic solutions and display specifications impact the perceived quality of the final output can be beneficial in creating automated and efficient pipelines that deliver high quality user experience. This knowledge is particularly important given that the issues mentioned earlier are just the tip of the iceberg, and many more artifacts can be encountered throughout the entire process of generating visual content.

Such knowledge can potentially be provided by perceptual models like image, mesh, and motion quality metrics. However, state-of-the-art methods have a handful of shortcomings that very often prevent them from being used in real-life applications. First of all, certain standard metrics, such as those based on simple formulas like Euclidean distance, are oblivious to how humans perceive visual stimuli. More complex solutions very often tend to focus on low-level features such as sharpness, contrast, or noise, which are more perceptually motivated but fail to reproduce *Human Visual System (HVS)* characteristics like *luminance masking* or *contrast masking* [Legge and Foley, 1980], which may reduce the visibility of distortions. As a result, these metrics may not always accurately reflect the visual quality of an image or the usability of a mesh.

Secondly, it is common to encounter domain-specific metrics, which may not be suitable for more general use. For example, a metric developed for evaluating the quality of medical images may not be appropriate for assessing the quality of images used in computer graphics. Similarly, metrics tailored to detect specific types of image or mesh artifacts, like noise or blur, may not be applicable when different artifact types are present or when they interact in unpredictable ways. This complexity can result in the metric either failing to detect distortions or providing an inadequate

measure of quality.

Furthermore, multiple state-of-the-art methods suffer from a lack of standardization. As a consequence, these metrics may yield arbitrary values that cannot be directly compared to the outputs of other metrics, as they operate on different scales or units. This inconsistency poses challenges in understanding the metric's score and comparing the results of various studies or evaluations of algorithm effectiveness.

Lastly, perceptual models typically require calibration or validation using subjective data. This can be a cumbersome and expensive process, as it often involves recruiting human subjects to participate in time-consuming studies. Additionally, collecting subjective data can be challenging because it requires careful selection and training of participants to ensure that they accurately reflect the characteristics of the target population and can provide reliable and consistent judgments of quality.

In this dissertation, our goal is to identify and address both the issues that arise at various stages of the computer graphics pipeline and the shortcomings of state-of-the-art metrics. With regard to the former one, we begin with an exploration of the image quality and visibility metrics, emphasizing the latter, as they can solve various image-related problems. Despite these metrics being primarily associated with the second stage of the graphics pipeline, we choose to introduce them first, as many of their foundational concepts and design choices are later utilized in mesh and motion quality assessments. We show that our neural-based solution yields superior accuracy, particularly in handling diverse computer graphics artifacts, allowing for automated assessment of a vast set of algorithmic solutions. Additionally, we suggest a method for fine-tuning this general-purpose metric to suit a specific task while minimizing the need for human subjects' involvement.

Next, we shift our focus to mesh quality assessment. Considering the significance of human faces as stimuli, to which we are highly sensitive, and their relevance in various applications, we assess the state-of-the-art mesh quality metrics in the context of distortions on meshes representing human faces. We propose crucial extensions of existing methods and propose applications of automatic LOD technique and perceptually lossless compression.

Finally, we address display-related problems and present a solution to achieve optimal visual quality for animated content within budgetary constraints. Our approach considers multiple factors, such as display specifications, motion velocity, luminance, contrast, and more. In addition, we investigate the issues arising when the content is displayed in a Virtual Reality setting, with emphasis on low luminance conditions.

In our pursuit of application-oriented solutions, we carefully address the challenges involved in designing perceptual models and the potential issues that may arise when using existing ones. Our approach encompasses several key elements.

Firstly, we prioritize proper standardization. The metrics we propose or modify are designed to deliver results in a format easily interpretable as a probability of distortion visibility, enhancing their usability and intuitive nature.

Secondly, our metrics are meticulously crafted with a deep understanding of human visual perception. This ensures that they are perceptually relevant and closely aligned with the characteristics of the Human Visual System.

We also contribute extensive datasets on the perception of image, mesh, and motion artifacts. This valuable data resource aids in overcoming challenges related to the calibration and validation of perceptual models that may be developed in the future.

Furthermore, we place a strong emphasis on the generalizability of our proposed metrics. This is achieved through the inclusion of a wide range of artifact types in our training datasets (visibility metrics) or simultaneous consideration of the most significant factors influencing the quality of moving content.

We hope that all described endeavors will contribute to the development of more efficient and automated designs for virtual experiences, which will have a positive impact on various fields such as entertainment, education, and beyond.

1.2 Contributions

In this thesis, we propose novel solutions in the fields of image, mesh, and motion quality that can simplify and automate multiple computer graphics applications. Additionally, we explore the impact of display design choices and parameters on the overall visual experience, striving for an end-to-end solution that delivers optimal results. All the proposed quality assessment metrics are linked to perception and offer a standardized, meaningful, and easily understandable quality evaluation.

Image Quality A large number of imaging and computer graphics applications require localized information on the visibility of image distortions. Existing image quality metrics are not suitable for this task as they provide a single quality value per image. Existing visibility metrics produce visual difference maps and are specifically designed for detecting just noticeable distortions, but their predictions are often inaccurate. Chapters 3 and 4 focus on delivering novel solutions that significantly reduce these limitations.

The main contributions of Chapter 3 (based on [Wolski et al., 2018; Ye et al., 2019]) are:

- the largest publicly available datasets (LocVis[1] and LocVisVC[2]) with local manual marking of visible distortions, arising in computer graphics and vision applications,
- a statistical inference model allowing robust fits to noisy subjective data,
- retraining of a number of popular metrics, which leads to their standardization and a significant improvement of their predictions,
- a CNN-based near-threshold visibility metric, which outperforms all existing metrics in a local prediction of distortion visibility,
- a hybrid model that combines white-box modeling of effects related to different peak luminance and viewing distance with a black-box deep neural network,
- a demonstration of the utility of visibility metrics provided through three practical applications: watermarking, visually lossless JPEG compression, and superresolution.

This chapter is an outcome of a collaborative effort between the Max Planck Institute for Informatics, the University of Cambridge, and the University College London. I hereby state that I am the main contributor to all the aforementioned aspects, with the exclusion of the hybrid model design and the two latter applications (compression

[1] LocVis: `https://doi.org/10.17863/CAM.21484`
[2] LocVisVC: `https://doi.org/10.17863/CAM.37996`

and superresolution). In order to ensure a comprehensive presentation, these parts have been included in this dissertation. However, it is crucial to acknowledge that although they are the result of collaborative work, the primary credit for these parts should be attributed to my collaborators.

One may postulate that to maximize the performance of a metric, it should be trained on a given task. In Chapter 4 (published as [Wolski et al., 2019]), following this idea, we:

- propose a strategy to fine-tune the CNN-based visibility metric using data from shorter and simpler user experiments, with an aim to improve the metric's performance,
- demonstrate the performance of such procedure in the context of texture optimization for game engines.

Mesh Quality To ensure high-quality human characters or avatars in gaming, entertainment, and social applications, it is important to reproduce their facial features accurately at the geometry level. While it is possible to measure mesh distortions objectively, it does not provide insights into how users perceive them. By quantifying the perceptual impact, geometry processing can achieve optimal outcomes by balancing cost and quality for key applications. In the domain of mesh quality evaluation Chapter 5 (result of an internship at *Meta Reality Labs*; published as [Wolski et al., 2022a]) contributes:

- a dataset of meshes representing human faces that follows the demographic distribution of the US census [2021], ensuring diversity,
- a large-scale perceptual database[3] of the visibility of geometric distortions on faces,
- attempt on standardization of the state-of-the-art metrics,
- an applications of our database, such as evaluating metrics' performance and setting perceptually meaningful thresholds for tasks like automatic mesh compression and LOD rendering.

Discrete Representation of Motion in Digital Displays To deliver a high-quality visual experience, it is crucial to account for the dynamic nature of displayed content. The motion of the visual stimulus can be a double-edged sword, as it may, on one hand, mask the spatial artifacts but also generate artifacts in the temporal domain. In cases where rendering resources are limited (e.g., time or power), identifying the optimal combination of rendering parameters like resolution and refresh rate is necessary for achieving the best quality of dynamic content. In Chapter 6 (based on [Jindal et al., 2021]) we tackle these issues with following contributions:

- Content-adaptive Metric of Judder, Aliasing, and Blur (CaMoJAB), derived from psychophysical models and calibrated on several datasets.
- a dataset[4] of motion quality for animations rendered with different shading rates, motion velocities, textures, refresh rates and display angular resolution (in pixels-per-degree), and persistence.

[3]Mesh distortion dataset: `https://github.com/facebookresearch/Geo-metric`
[4]CaMoJAB: `https://www.cl.cam.ac.uk/research/rainbow/projects/alsarr`

- Adaptive Local Shading and Refresh Rate (ALSaRR) method for control of real-time rendering, which maximizes animation quality under a limited budget.

This chapter results from a collaboration between the Max Planck Institute for Informatics and the University of Cambridge. I hereby state that I am the main contributor in the latter aspect, which included the GPU implementation of the method.

Scotopic and Photopic Luminance Levels in Stereoscopic Displays Displaying VR content at low brightness is sometimes necessary or desirable due to factors such as reducing flicker artifacts or prolonging battery life. However, displaying content at low luminance can decrease the reliability of binocular depth cues, which can affect the viewer's experience. To solve this issue, in Chapter 7 (published as [Wolski et al., 2022b]) we contribute:

- psychophysical measurements of the effect of luminance on our ability to infer 3D shapes from binocular disparity cues,
- a model of stereoscopic constancy,
- a contrast enhancement algorithm that improves depth perception on dimmed displays, with applications to VR and 3D movies,
- an implementation of the algorithm as an efficient post-processing GPU shader, which can be used directly in the Unity Engine[5].

This chapter has emerged from a collaborative endeavor involving the Max Planck Institute for Informatics and the University of Cambridge. I hereby state that I am the main contributor to all the above-mentioned aspects, while the shared contribution can be attributed to the model of stereoscopic constancy.

1.3 Outline

The thesis is structured as follows. Chapter 2 provides an overview of relevant theoretical background. In Chapter 3, we introduce a novel CNN-based visibility metric, which we further extend by accounting for varying viewing conditions, namely viewing distance and display luminance. We demonstrate the applicability of such a metric in the local perceptually lossless compression, superresolution, and watermarking tasks. Next, in Chapter 4, we propose guidelines on how a general purpose metric can be fine-tuned for a specific task and use outcomes of such an approach to create a tool for automatic texture size optimization in a game engine. In Chapter 5, we discuss mesh quality in the context of geometry representing human faces, propose a new dataset of faces, and demonstrate a handful of its applications. Then, in Chapter 6 we move to the motion quality domain. We investigate the relation between both static and dynamic characteristics of displayed content and display parameters. We conclude this investigation in the form of a Content-adaptive Metric of Judder, Aliasing, and Blur. In Chapter 7, we look into how dimming the VR display might impact user experience and develop the stereoscopic constancy model that can be used to eliminate shortcomings of such an approach. Finally, Chapter 8 concludes the thesis and discusses potential future directions.

[5]Stereo constancy model: `https://dark-stereo.mpi-inf.mpg.de`

Chapter 2

Background and Previous Work

In this chapter, we provide a review of concepts, tools, and previous work relevant to this thesis. First, we recollect fundamental knowledge about the human visual system (Section 2.1). Further, we discuss principles and applications of mesh, image, and motion quality metrics in Section 2.2. Finally, we shortly describe the technical aspects and tools used in this dissertation (Section 2.3).

2.1 Human Visual System

The HVS is a complex and remarkable biological system that allows us to perceive and make sense of the visual world around us. From the moment light enters our eyes, it is transformed and processed by a network of specialized cells and neural circuits that ultimately give rise to our visual perception. In this section, we will take a closer look at the anatomy of the HVS (Section 2.1.1) and biological mechanisms underlying several important aspects of human vision, including contrast perception (Section 2.1.2), motion perception (Section 2.1.3), and depth perception (Section 2.1.4). We will also explore how the visual system behaves in low luminance conditions (Section 2.1.5) and take a look into human visual system phenomena that make us insensitive to certain types of visual stimuli (Section 2.1.6).

2.1.1 Anatomy of Human Visual System

This section covers the anatomy of the human visual system, starting with a description of the eye, followed by an explanation of the phototransduction process, and concluding with insights into the processing that occurs in the primary visual cortex.

Anatomy and Function of the Eye

The human eye is a complex structure consisting of multiple components, each with its own unique purpose. Here we focus on the most relevat optics-related components on the light path. Positioned at the front of the eye, the *cornea* is a transparent and curved layer that serves as a barrier that protects the inner eye. Additionally, the cornea plays a crucial role in refracting incoming light to ensure it is properly directed onto the delicate photosensitive layer that covers the rear of the eye, called *retina*.

Located behind the cornea is the *iris*, the pigmented portion of the eye. Comprising two sets of muscles that collaborate, the iris modifies the size of the *pupil* (the opening that allows light to enter the eye) in response to changes in light intensity or emotional arousal. When the lighting is dim, the pupil expands to allow more light to pass through, while in bright conditions, it contracts to restrict the amount of light that enters the eye.

Subsequently, the light passes through the *aqueous humour*, a water-like fluid, and the *lens*, a transparent component that sits posterior to the iris. Through the agency of the connected muscles, the lens can alter its shape to concentrate the light rays reflecting off objects situated at varying distances. This process is called *accommodation* and is necessary to sharpen the focus of the light on the retina. It is noteworthy that the lens, in conjunction with the cornea, is an imperfect optical instrument that not only concentrates the light but also distorts the retinal image, introducing chromatic aberration and blurring. These optical imperfections can limit the resolution of human vision by reducing the sharpness and clarity of the image [Walls, 1944; Wandell, 1995].

The retina, which was mentioned earlier, consists of millions of light-sensitive cells known as *photoreceptors*. Its main function is to convert the incoming light into neural signals that are then sent to the brain for processing. There are two types of photoreceptors in the retina, namely *rods* and *cones*. Rods are highly sensitive to low levels of light and are responsible for vision in dim lighting conditions. They can even respond to the absorption of a single photon [Baylor et al., 1984]. Conversely, cones are responsible for color vision and are most sensitive to bright light. There are three types of cones, named S, M, and L, each with peak sensitivity to a specific range of light wavelengths that correspond to the colors blue, green, and red.

The distribution of cones and rods in the retina is not random; it is highly organized. The central region of the retina features a small circular indentation, known as *fovea*, that is approximately 1.5 mm in diameter. The fovea consists solely of cone cells, providing high spatial resolution, about 60 cycles per degree.

It is noteworthy that the arrangement of the three types of cones in the fovea is also highly organized. Due to the chromatic aberration introduced by the lens, not all wavelengths of light can be focused simultaneously. The wavelengths that M- and L-cones are sensitive to are close to each other and are less susceptible to optical imperfections, making them a priority for the eye's accommodation process. Consequently, S-cones are less densely packed than M- and L-cones, since blue light is not sharply focused on the retina due to chromatic aberration, and a high density of S-cones would lead to the detection of a blurred signal [Roorda and Williams, 1999].

As we move away from the fovea, the number of cones decreases, and the number of rods increases. The density of rods reaches its highest at 10 to 20 degrees away from the fovea, and this decrease in cone density leads to a reduction in spatial resolution with increasing eccentricity [Curcio et al., 1990].

The visual overview of the eye anatomy is presented in Figure 2.1.

Phototransduction and Pathway to the Brain

To better understand the loss of resolution with eccentricity, one needs to look closer at how the visual signal is processed down the visual pathways.

When light is absorbed by a photoreceptor cell, a cascade of biochemical reactions, known as *phototransduction*, is triggered. This process involves the activation of a series of proteins that ultimately leads to the release of a neurotransmitter that signals to the next neuron in the visual pathway [Wandell, 1995].

The visual pathway is a complex network of neurons that extends from the retina to the brain. The first neurons which receive input from the photoreceptor cells are called *bipolar cells*. The bipolar cells act as an intermediary between the photoreceptor cells and the *ganglion cells*, processing and filtering the visual information. There are two types of bipolar cells: ON cells and OFF cells. ON cells are excited by an increase in light intensity, while OFF cells are excited by a decrease in light intensity,

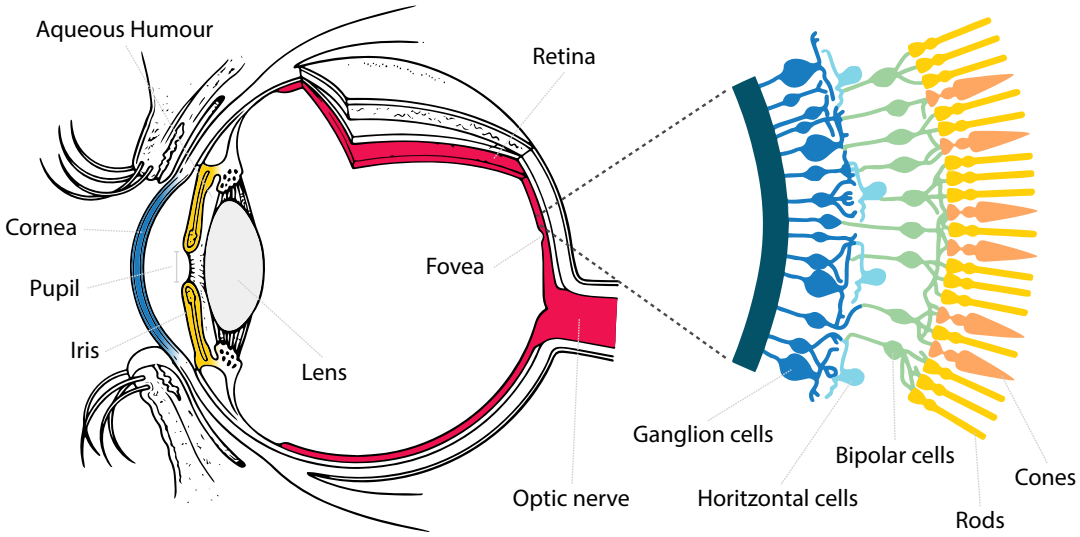


FIGURE 2.1: *Anatomy of the eye and magnified preview of the layered structure of the retina.*

in effect amplifying or reducing the signal transmitted from photoreceptors [Hoshi et al., 2009].

The next layer of neurons, called ganglion cells, receives signals from the bipolar cells in the retina. While there are several types of ganglion cells in the eye, this overview will focus on the two most common types that have the biggest impact on viewing resolution: *midget ganglion cells*, which are also known as parvocellular (P) cells and *parasol ganglion cells*, also called magnocellular (M) cells. These cells differ in the size of their dendritic fields and the number of connections to the photoreceptor cells, which determines their function. Midget cells in the fovea receive input from a single cone and are responsible for processing color and fine details in the center of the visual field. Parasol cells, on the other hand, are larger and converge information from multiple photoreceptors, resulting in lower resolution response. They are responsible for detecting motion, contrast, and changes in overall brightness. Due to the larger size of their dendritic fields, fewer parasol cells are needed to cover the whole retinal area. It has also been observed that the size of dendritic fields of all ganglion cells increases with the eccentricity, causing loss of resolution in the peripheries [Dacey and Petersen, 1990; Watanabe and Rodieck, 1989]. In addition to bipolar and ganglion cells, the retina contains several other types of cells that play important roles in processing visual information, including *horizontal cells*. These cells are responsible for adjusting the sensitivity and response characteristics of the photoreceptor cells [Wässle et al., 1990].

Finally, the axons of the ganglion cells exit the retina at the *optic disc* of the eye and travel through the *optic nerve*, which is composed of over a million nerve fibers. The optic nerve fibers from each eye partially cross over at the *optic chiasm* and then project to the *lateral geniculate nucleus (LGN)* in the thalamus. The LGN has six layers, and each layer receives input from different types of ganglion cells, providing parallel processing of visual information. From the LGN, visual information is transmitted to different areas of the located in the occipital lobe *primary visual cortex (V1)*, also known as the striate cortex or the visual cortex [Wandell, 1995].

Primary Visual Cortex and Beyond

The visual cortex serves as the primary processor of visual information and comprises over a dozen (estimated as approximately 15-20 [Van Essen and Maunsell, 1983]) distinct areas categorized by structural and functional criteria. It is hypothesized that with each subsequent cortical area, visual information becomes more specialized. Neurons in the visual cortex react to stimuli within a particular *receptive field*, which denotes the visual field area they respond to, and neurons in each visual area have unique response patterns to varying stimuli (more information on receptive fields can be found in Section 2.1.2). Among the extensively studied specialized cells are simple and complex cells. Simple cells, primarily located in V1, respond to specific visual cues such as edge orientation. Conversely, complex cells in V1-V3 respond to summations of receptive fields from several simple cells and mainly respond to movement in specific directions [Goodale and Milner, 1992].

V1 is the initial cortical region to receive and process visual information and is the most extensively studied segment of the visual cortex. It consists of six distinctive layers. These layers are numbered from 1 to 6. Each layer contains different types of neurons with unique connectivity patterns and functions. Notably, layer 4 is where the lateral geniculate transmits information to V1, and it contains the highest concentration of simple cells. Complex cells, however, are present in layers 2, 3, and 6. V1 is responsive to uncomplicated visual elements, including orientation and direction of a stimulus, and this input is fundamental for more intricate pattern recognition further along the visual pathway [DeAngelis et al., 1995]. It is also worth noting that V1 is the brain region that receives and integrates information from both eyes, thus allowing for binocular vision, depth perception, and the ability to perceive the world in three dimensions.

After receiving integrated information from V1, V2 exhibits an enhanced level of complexity and response patterns to objects. Scientists have observed cells in this region reacting to variations in color, spatial frequency, moderately intricate patterns, and object orientation [Roe and Ts'o, 1995; Livingstone and Hubel, 1988; Tanaka, 1996]. V2 sends feedback connections to V1 and has forward connections with V3-V5. The information exiting the second visual area divides into the dorsal and ventral streams, which specialize in processing distinct visual information aspects. The former is frequently associated with object recognition, while the latter concentrates on spatial tasks and visual-motor abilities.

As visual information travels through the brain, more specialized cells emerge. It is believed that certain cells or clusters of cells become specialized in reacting to specific objects, enabling the immediate recognition of previously encountered items [Quiroga, 2012].

2.1.2 Contrast Perception

Contrast perception is a fundamental aspect of visual processing that allows us to sense relative differences in luminance. In this section, we will explore the basic mechanisms of contrast perception, including the role of receptive fields in detecting contrast. We will then focus on the contrast sensitivity function, which describes the neural response to contrast as a function of spatial frequency. We will discuss how this function can be used to characterize the properties of single neurons and how it relates to the overall contrast sensitivity of the human visual system as a whole.

Mechanism of Contrast Perception

To understand the mechanism of contrast perception, we need to go back to the retina of the eye and the located there bipolar cells. As mentioned before, there are two types of bipolar cells, ON and OFF, which respond to changes in light in opposite ways. ON bipolar cells are excited by light and produce an increase in activity, while OFF bipolar cells are inhibited by light and produce a decrease in activity [Nelson, 1973].

The outputs of ON and OFF bipolar cells are combined in retinal ganglion cells to create the center-surround receptive field. Ganglion cells also come in two types: ON-center and OFF-center cells. Specifically, in the case of the ON-center ganglion cell, signals from ON bipolar cells excite the central region of the receptive field of the retinal ganglion cell, while signals from OFF bipolar cells inhibit the surrounding region. The OFF-center ganglion cells, on the other hand, are inhibited in the central region, while the surround is excited by the signal from ON bipolar cells. It is worth noting that the area of dendritic fields of ganglion cells varies significantly; hence different ganglion cells will have receptive fields of different sizes. The ON- and OFF-center receptive fields are illustrated in Figure 2.2.

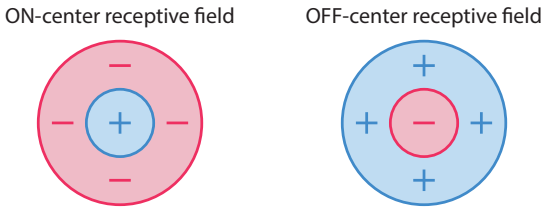


FIGURE 2.2: *Illustration of the ON-center and OFF-center receptive fields. The + sign and blue color denote the area of excitation, while the − sign and pink color denote the inhibition area.*

The final step in contrast perception occurs in the visual cortex, where the brain processes visual information received from the eyes. Neurons in the primary visual cortex pool have more complex receptive fields than those in the retina, which allows them to detect more complex visual patterns. For example, neurons in the primary visual cortex might respond to specific orientations of edges (orientation selectivity), summing the responses of neurons with non-oriented, circularly symmetric receptive fields of the ganglion cells [Hubel and Wiesel, 1962]. Such summation and the concept of orientation-selective receptive fields are presented in Figure 2.3.

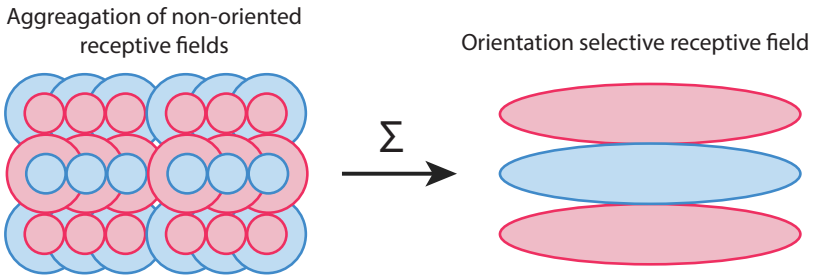


FIGURE 2.3: *Creation of orientation selective linear receptive fields. The presented receptive field excites when it is aligned with a horizontal line and inhibits when the vertical stimulus is present.*

As one can imagine, the further down the visual pathways, the more complex and specialized receptive fields develop, which are important for more advanced visual processing tasks such as object recognition and others.

Contrast Sensitivity Function

In neuroscience, a common practice involves studying how neurons respond to harmonic functions. This approach can be utilized to characterize a neuron's *contrast sensitivity function (CSF)*, which describes the relationship between the contrast of a stimulus and its spatial frequency (Figure 2.4). CSF also provides information regarding the neuron's spatial receptive field. The dominant spatial frequency of the neuron's response indicates the overall size of its receptive field, while the decrease in sensitivity at low spatial frequencies provides information about the strength of the inhibitory surround. Furthermore, the decrease in sensitivity at high spatial frequencies explains the size of the receptive field center. Neurons that possess small centers are more responsive to high spatial frequency targets, whereas those with large centers exhibit less responsiveness. Understanding the contrast sensitivity function and its relationship to the receptive field can provide insights into the visual processing capabilities of neurons in the visual system.

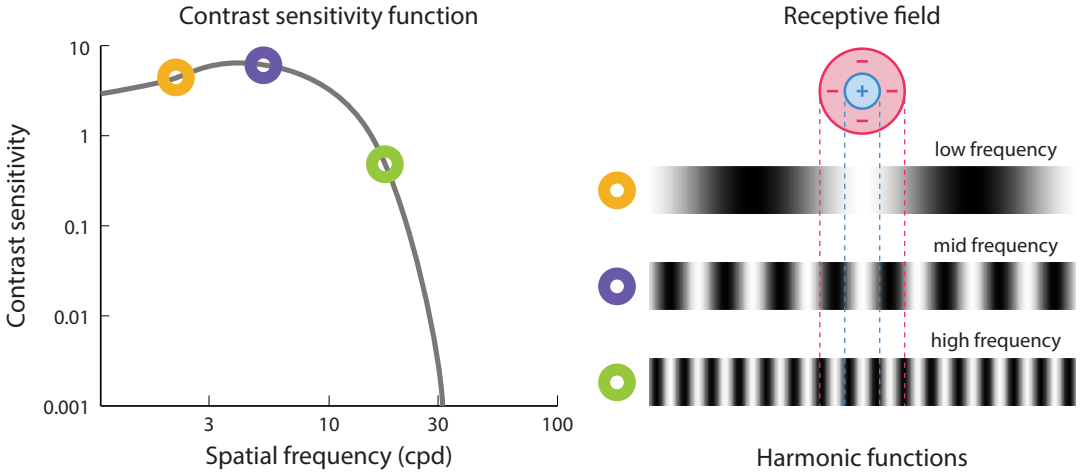


FIGURE 2.4: *Contrast sensitivity of a linear on-center neuron. The illustrated example shows that a neuron exhibits optimal response to a medium spatial frequency harmonic function, where bright bars stimulate the on-center and dark bars stimulate the surround (purple dot). In contrast, low spatial frequencies cause signals from the center and surround to counteract each other, reducing sensitivity (yellow dot). Even more extreme reduction of sensitivity can be observed for high spatial frequencies, which are averaged over the center of the receptive field (green dot). The figure has been drawn based on [Wandell, 1995].*

CSF is useful not only for characterizing the response properties of individual neurons. In psychophysics, it is used to describe the response of the human visual system as a whole. By measuring the CSF of human observers, it is possible to assess the overall quality of the visual system and identify any deficiencies or limitations in visual function. The CSF is particularly useful in this context because it provides information about the visual system's sensitivity to different spatial frequencies, which is important for many visual tasks, such as detecting small details or discriminating between different objects. As we learn later in Section 2.2, such a form of contrast sensitivity function is commonly used in quality metrics to predict how distortion introduced to visual signal impacts its perceived quality. Multiple CSF models have been developed and widely used to study visual perception [Campbell and Robson, 1968; Barten, 1999; Barten, 2003b; Watson and Ahumada, 2005; Pelli and Bex, 2013; Mantiuk et al., 2022].

Overall, the contrast sensitivity function is a versatile and powerful tool for understanding the response properties of neurons in the visual system and assessing the quality of the human visual system as a whole.

2.1.3 Motion Perception

Motion perception is a critical aspect of visual processing that allows us to detect and track moving objects in our environment. In this section, we will explore the basic mechanisms of motion perception, including the role of spatio-temporal receptive fields in detecting motion. Then, we will examine motion artifacts that can arise in the context of modern display technology, such as motion blur and judder, and discuss how they can impact our perception of motion. Finally, we will discuss factors that might affect the perception of motion artifacts.

Mechanism of Motion Perception

Neurons in the visual system are not only sensitive to changes in contrast and frequency but also exhibit varying responses to stimuli over time. These neurons have been observed in various areas of the visual system, including the primary visual cortex and higher visual areas, and are thought to play a role in motion perception and object tracking. In order to understand how the visual system processes dynamic stimuli, researchers have developed the concept of space-time receptive fields. These are receptive fields that combine information about spatial and temporal features of visual stimuli, allowing neurons to respond selectively to stimuli that exhibit specific spatio-temporal patterns.

To illustrate the concept of space-time receptive fields, we can consider the example of a vertical bar moving from left to right over time, presented in Figure 2.5. On the left-hand side, we can see the motion sequence presented in a three-dimensional space-time coordinate system. Given a single moment of time, we can project such volume to two-dimensional space (x, y), representing what the human eye sees spatially (central part of Figure 2.5). As the stimulus is spatially one dimensional (no changes in y axis), it can be projected to the spatio-temporal dimension (t, x) presented on the right-hand side of Figure 2.5.

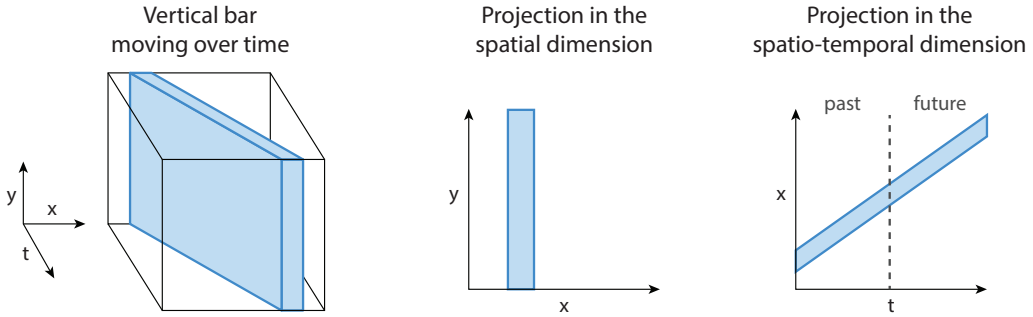


FIGURE 2.5: *The motion sequence of vertical bar presented as a three-dimensional volume (left), its projection to spatial dimension (x, y) given a single moment of time (center), and projection to spatio-temporal dimension (t, x) (right). The figure has been drawn based on [Wandell, 1995].*

The orientation of the stimulus in the spatio-temporal projection is highly influenced by the velocity of the bar. If the bar is not moving, the orientation will be horizontal, whereas, with an increase of velocity, the orientation changes to a vertical position [Adelson and Bergen, 1985; Watson and Ahumada, 1985]. This orientation is

crucial in developing space-time receptive fields, which are depicted on the left-hand side of Figure 2.6. As events from the recent past stimulate the neuron, the receptive field shifts along the time axis to stay in sync with the current moment. Similarly to the concept described in Section 2.1.2, the inhibitory and excitatory regions are represented by pink and blue areas, respectively.

The central and right parts of Figure 2.6 illustrate the response of the space-time receptive field to aligned and non-aligned stimuli, respectively. In both cases, the upper part of the graph displays a plot of a moving line (light gray bar) in (t, x) coordinates and the space-time receptive field of a linear neuron. The neuron's receptive field is depicted at different moments in time, indicated by vertical dashed lines. A large amplitude response is produced when the space-time receptive field's orientation matches that of the stimulus motion, as shown in the center part of the figure. However, when the stimulus motion is not aligned with the space-time receptive field, as demonstrated on the right side of Figure 2.6, the response amplitude is much smaller. To summarize, neurons with space-time receptive fields tuned to a given type of motion would respond more strongly to this specific pattern of visual input than to other patterns, such as a stationary bar or a bar moving in a different direction.

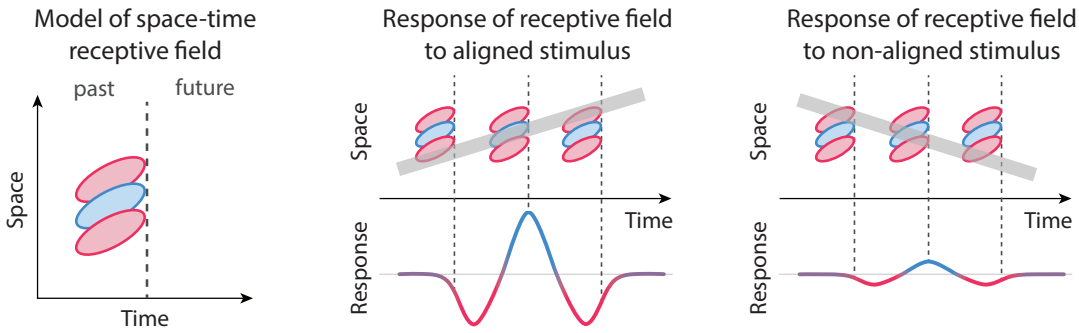


FIGURE 2.6: *Model of space-time receptive field (left). The central and right parts of the figure present the response of the space-time receptive field to aligned and non-aligned stimuli in the spatio-temporal domain. The upper part of the plots displays a moving line in (t, x) coordinates, as well as the space-time receptive field of a linear neuron. A significant response is generated when the orientation of the stimulus motion matches that of the space-time receptive field. The figure has been drawn based on [Wandell, 1995].*

Motion Artifacts on Modern Displays

Real-time rendering of animated content is prone to various artifacts due to software and hardware limitations. Working with frame rates lower than display refresh rates can give rise to screen *tearing* or *stuttering* artifacts. Asymmetric pixel transition time of LCD panels leads to *trailing* artifacts. Response time compensation used to mitigate trailing may lead to *coronas (glowing edges)* [Jokinen and Nivala, 2017]. *Blur* is perceived whenever a tracked object has a non-zero retinal velocity. This can happen either due to the sample and hold nature of some display technologies (hold-type blur) or imperfect eye motion when tracking an object [Denes et al., 2020]. One way to reduce blur is to reduce the amount of time a signal is displayed every frame (aka persistence), but this can lead to *flicker* if the display's refresh rate is lower than the critical fusion frequency of the human visual system (HVS) and also from a lower accuracy of saccadic eye motion [Goettker et al., 2020]. Low refresh rates also lead to *judder*/non-continuous motion perception. On low persistence displays, such as

HMDs or projectors with butterfly shutters, *ghosting/false-edges* is a common artifact that occurs when the image of a moving object is displayed multiple times at the same location [Murdison et al., 2019]. Finally, the visibility of these artifacts also depends on the displayed content. When the display's spatial and temporal sampling frequency is lower than that of the displayed signal, we see *aliasing* artifacts which are a common occurrence in real-time graphics.

The artifact visibility can be suppressed by actively modifying image content. For example, increasing motion blur by means of rendering [Sung et al., 2002; Navarro et al., 2011] or longer camera exposure [Fuchs et al., 2010] can reduce aliasing, ghosting, and flicker [Stengel et al., 2015; Hoffman et al., 2011]. Artifact suppression can also be achieved by hardware adjustments such as syncing display refresh rate with the frame rate of an application (adaptive-sync) or over-driving liquid crystals to reduce their response time [Feng, 2006]. Unfortunately, achieving an artifact-free motion rendering is often an impossible task due to the current display limitations and usually involves trading off between different artifacts. In the next part, we discuss several factors that affect the perception of these artifacts and motion quality.

Factors Affecting Perception of Motion Artifacts on Modern Displays

The perceived quality of reproduced motion is a well-studied problem, with various works exploring the effect of the following display and content-dependent factors on the quality of motion:

Refresh-Rate and Velocity: There is a consensus among multiple studies examining a wide range of refresh rates and velocities (refer to Table 2.1) that higher refresh rates are required to maintain the quality at higher velocities of motion. Mackin et al. [2016] report that even for a simple stimulus, it can take up to 600 Hz (depending upon the velocity) to achieve an artifact-free motion. However, observers found the motion artifacts beyond 300 Hz to be tolerable. They also noticed that temporal aliasing artifacts (judder and ghosting) contribute more to motion quality impairment compared to hold-type blur. In another study, Kuroki et al. [2007] measured motion quality up to a refresh rate of 480 Hz and found that the quality improves rapidly with increasing refresh rate but saturates after 240 Hz for natural images. Other studies, including those by [Wilcox et al., 2015; Chapiro et al., 2019a; Denes et al., 2020] have reported similar trends. The visibility of flicker, much like temporal aliasing, decreases with an increase in the refresh rate. These trends stay mostly the same for stereoscopic presentation [Hoffman et al., 2011]. However, unlike temporal aliasing, flicker can be detectable up to 500 Hz during saccadic eye motion [Davis et al., 2015].

Persistence: Also known as duty-ratio or shutter angle, is the fraction of the frame duration the signal is displayed. A substantial difference in judderness can be perceived [Daly et al., 2015] between shutter angles that differ by a large amount. Furthermore, it is necessary to lower persistence to significantly reduce the hold-type blur inherent to LCDs, as simply increasing the refresh rate is not effective [Sluyterman, 2006]. In fact, LCD panels with strobing backlights [Feng, 2006] are becoming increasingly common in head mounted displays (HMDs) and gaming monitors [Rejhon, 2017]. Unfortunately, psychophysical studies on the range of persistence used in these devices are quite limited (refer to Table 2.1).

Luminance and Contrast: The models of contrast sensitivity [Barten, 2003a] predict that the HVS is more sensitive to higher luminance and contrast and thus one would

expect a similar trend for visibility of motion artifacts. Larimer et al. [2001] verified this in their study, where they measured the effect of luminance, contrast, and refresh rate on judder and flicker up to 100 cd/m^2 and found luminance to be the dominant variable. Roberts and Wilkins [2013] noted that the visibility of flicker on a square-wave signal increases with increasing Michelson contrast and Daly et al. [2015] reported similar degradation of overall motion quality with increasing contrast.

Resolution and Spatial Frequency: Visibility of blur is linearly dependent on the logarithm of the averaged spatial frequency of an image [Kuroki et al., 2007] with observers showing a slightly higher preference for higher spatial frequency content [Navarro et al., 2011; Daly et al., 2015]. The content resolution has also shown to be an important factor for rendering on variable refresh rate displays, with viewers picking higher refresh rates for low computational budgets and higher resolution for higher budgets [Debattista et al., 2018]. Reducing the resolution in rendering results in a blur and aliasing artifacts, but how our sensitivity to these artifacts changes with velocity is still not well studied (Table 2.1).

Other Factors While there is also some evidence of an effect of motion direction [Daly et al., 2015], motion predictability [Denes et al., 2020], color [Daly et al., 2015] and the type of eye motion [Roberts and Wilkins, 2013] on motion quality, we do not explore them in this dissertation, and focus only on the previously-mentioned factors.

We summarize the above discussion in Table 2.1 and report the range of each parameter measured in the existing work. We find the studies of persistence and resolution to be particularly lacking as they do not cover the ranges commonly found in VR and mobile devices. Furthermore, spatio-temporal aliasing artifacts due to low-resolution rendering are also not well considered in these works. Lastly, a majority of these studies used oversimplified stimuli, such as lines, boxes, etc., that may not generalize well to complex image content. We attempt to fill these gaps in knowledge by conducting a psychophysical study on multiple display setups and more realistic texture content as described in Section 6.3.3. In the first column of Table 2.1, we mark all datasets, including ours, that we used to calibrate and validate our motion quality model (Section 6.3).

2.1.4 Depth Perception

Depth perception is critical for our daily lives as it allows us to navigate and interact with the world around us. Accurate depth perception is necessary for tasks such as reaching for objects, walking through a crowded environment, and driving a car. Without depth perception, we would struggle to accurately judge distances, leading to potentially dangerous situations.

In this section, we will briefly categorize and describe the depth cues. Next, we will focus on the stereopsis mechanism and its biological background. Lastly, we will explore the process of achieving the stereo vision in a virtual reality setup and examine the latest advancements aimed at enhancing the stereo vision experience in VR.

Stereo Vision Cues

Stereo vision cues are essential for depth perception and accurate spatial awareness. Our brain uses these cues to construct a three-dimensional representation of the

TABLE 2.1: *Existing motion quality studies and the range of motion artifacts and display/content parameters explored by them. Our experiments focused on understanding the effect of resolution and display persistence on visibility of motion artifacts, particularly blurring and aliasing artifacts caused by variable-rate shading (VRS).*

Motion Quality Datasets	Motion Artifacts						Range of Display/Content Parameters Studied						
	Judder	Flicker	Blur	Ghosting	Aliasing (Rendering)	Depth Distortion	Refresh Rate (Hz)	Persistence* (Duty Cycle)	Resolution (PPD)	Velocity (deg/s)	Adaptation Luminance (cd/m2)	Contrast (Michelson)	Stimuli
[Larimer et al., 2001]	✓	✓		✓			15 : 30	0.01 : 1	✗	0.46 : 5.46	7 : 110	0.5 : 1	Vertical Bars
[Kuroki et al., 2007]	✓		✓				60 : 480	CRT	64	0 : 80	40	Images	CG+Natural Images
[Navarro et al., 2011]			✓		✓		60	1	110	5 : 15	300	✗	CG Images
[Hoffman et al., 2011]	✓	✓	✓	✓		✓	10 : 50	CRT	42	1 : 16	30 : 60	✗	Moving Box
[Roberts and Wilkins, 2013]		✓					1000 : 5000	0.01	✗	0	0.02 : 310	0.05 : 0.4	Stationary Lines
[Johnson et al., 2014]	✓	✓	✓	✓		✓	30 : 240	0.25 : 0.5	60	0 : 25	10 : 50	✗	Moving Box
[Davis et al., 2015]		✓					25 : 1000	✗	87	0	20 : 2700	0.09 : 0.95	Stationary Edge
[Wilcox et al., 2015]	✓		✓				24 : 60	DLP	118	Videos	16	Videos	Natural Videos
[Daly et al., 2015]	✓	✓	✓	✓			12 : 60	DLP	30	0.8 : 1.3	5 : 65	0.125 : 1	Gabor + Natural Images
[Mackin et al., 2016]§	✓		✓	✓			60 : 2000	0.01	Printed Stimulus	10 : 70	150 : 3200	✗	Moving Edge
[Debattista et al., 2018]	✓		✓		✓		15 : 120	1	640 : 2560 px	3D Animation	350	Images	CG Videos
[Denes et al., 2020]§	✓		✓				15 : 165	1	62	15 : 45	36	Images	CG+Natural Images
ITU-R [Series, 2020]§	✓	✓	✓	✓			60 : 240	0.3 : 0.9	7 : 60	8 : 32	169 : 480	Videos	Natural Videos
Ours[Jindal et al., 2021]§	✓		✓		✓		60 : 144	0.05 : 1	13 : 90	0 : 75	2.5 : 150	0.7 : 0.9	CG+Natural Images

* Persistence is the display's duty cycle and should not be confused with the camera's shutter angle. It is not well-defined for CRT and DLP displays due to asymmetric phosphorus decay time and use of color wheel, respectively.
✗ Not reported in their publication.
§ Datasets that were used to calibrate/validate our model.

world around us, which is necessary for tasks such as navigation, object recognition, and depth estimation.

There are two main categories of stereo vision cues: monocular and binocular cues. Monocular cues are perceptible with a single eye and include cues such as perspective, shading, texture gradient, occlusion, and motion parallax. These cues prove valuable in providing important information about depth and distance in scenarios where only one eye is functional, such as in the case of vision loss in one eye, but they are limited in their accuracy and reliability.

On the other hand, binocular cues require both eyes and are more reliable and allow for faster and more accurate depth perception [Mikkola et al., 2010]. One of the binocular cues is *binocular disparity*, which arises from the differences in the retinal position of the same object in the two images received by the left and right eyes. Another binocular cue is *convergence*, which arises from the angle between the two eyes required to focus on a particular object [Julesz, 1964]. When an object is closer, the eyes must converge more to maintain a fusion of left and right images, providing a cue for depth perception. Due to the importance of the binocular cues in the mechanism of stereo vision, they are the main focus of Chapter 7.

Stereopsis

Stereopsis is the visual perception of depth and 3D structure that is made possible by the fact that our two eyes are located apart from each other, providing a slightly different view of the same scene. This difference in perspective creates a disparity between the two images. Such information is used in the process of binocular fusion, which occurs in the visual cortex of the brain, to create a sense of depth.

In order to gain a better understanding of this mechanism, one shall examine Figure 2.7. The figure depicts a situation where two eyes are positioned in a way that allows the blue point to project an image onto both foveae. This indicates that the images of the point fall on matching retinal locations and thus have zero *retinal disparity*. The collection of points with zero retinal disparity forms an arc around the viewer known as the *horopter*.

If a point is located nearer or farther away than the horopter, its image will project onto non-matching retinal positions. The difference between the image locations and their corresponding positions is referred to as *retinal disparity*. As the primary separation between the eyes is horizontal, the majority of retinal disparities are also horizontal in direction. When both images fall on the nasal side of the foveae, the object is located behind the horopter, while when the images fall on the temporal side, they are closer to the observer than the horopter. This mechanism allows us to assess the depth of the objects in the environment.

It is worth noting that the human visual system can merge images of objects located on the horopter and within a narrow range surrounding it known as *Panum's fusional area*, providing an impression of single vision. However, when an object falls outside the boundaries of Panum's fusional area (gray points in Figure 2.7), either in front of or behind it, the visual system struggles to correctly merge the images, resulting in double vision, also known as diplopia.

Researchers attempted to identify how depth information is coded by neurons in our brains. One possibility is to create a group of neurons, each of which responds to only a single disparity, and to estimate local disparity by identifying the neuron with the largest response. Alternatively, Richards [1971] suggested measuring disparity by creating a few pools of neurons with coarse disparity tuning. One pool could consist of neurons that respond when an object feature is beyond the horopter, another of

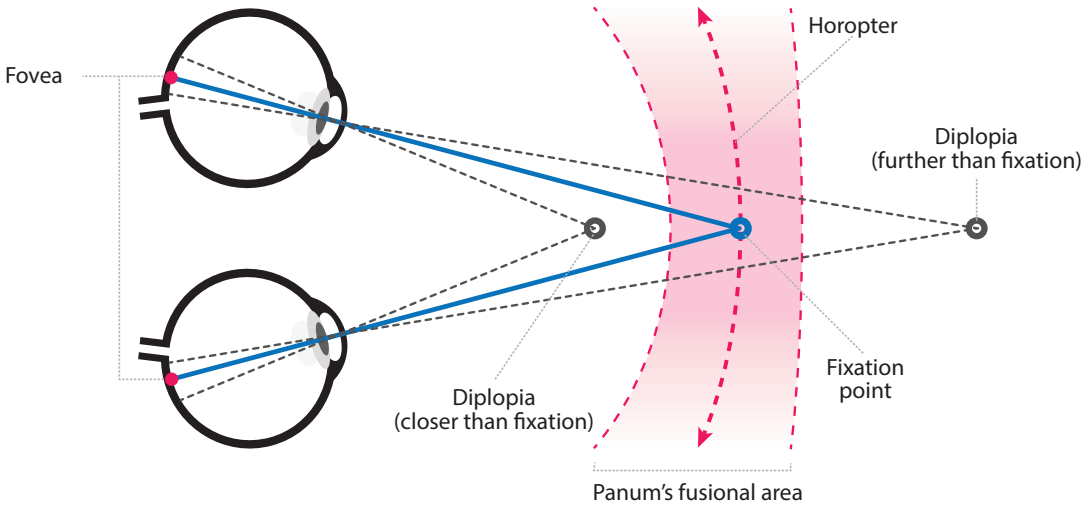


FIGURE 2.7: *Mechanism of stereopsis. Points that project images onto corresponding retinal positions create the horopter, resulting in zero retinal disparity. Points located outside the horopter exhibit a retinal disparity greater than zero, providing information about the relative distance to the focal point. However, when a point falls beyond the Panum's fusional area, the retinal disparity becomes too large, preventing fusion and leading to diplopia (double vision).*

neurons that respond when the feature is in front of it, and a third of neurons that respond only when the feature is close to the horopter. To estimate depth, one would compare the relative responses in the three neural pools.

Poggio and Talbot [1981] measured how well individual neurons responded to stimuli with different amounts of disparity and confirmed Richards' hypothesis. They discovered four types of neurons that responded differently to the retinal disparity. The first two neurons responded to disparities near the fixation plane, and stimuli near the horopter stimulated or inhibited these cells. The other type of neuron was excited by a stimulus with a disparity that placed the object beyond the horopter, and it was inhibited by stimuli in front of the horopter. The last type of neuron exhibited an approximately complementary excitation pattern. The neuron was excited by objects closer than the horopter and inhibited by objects further away. These measurements support Richards' hypothesis that binocular depth is coded based on the response of neurons organized in disparity pools.

Reproducing Stereo Vision in VR

In VR, stereopsis is recreated using stereoscopic rendering. This technique, similar to the real-world phenomenon, involves displaying two slightly different images of the same scene, one for each eye, to create the illusion of depth perception. This is achieved using two separate displays, each rendering a scene from a slightly different position, corresponding to the user's eyes position. The displays are arranged to create a binocular overlap zone, where the two images overlap, creating a three-dimensional image.

While the VR system can accurately present different images to each eye, allowing for the proper retinal disparity, the accommodation of the eyes is disrupted. Within the currently available VR products, the viewer's eyes typically focus on a fixed distance as the displayed image is presented on a screen. This mismatch between vergence (eye convergence) and accommodation (eye focusing) can lead to a phenomenon known as vergence accommodation conflict. The visual system becomes

confused as the eyes converge to perceive depth while the focal point remains fixed, resulting in visual discomfort, eye strain, and a less immersive experience.

Enhancements of Stereoscopic Experience

Disparity Manipulation Several works have proposed techniques for altering image disparity, mostly intending to reduce vergence-accommodation conflict, which can make images uncomfortable to view [Wann et al., 1995]. Oskam et al. [2011] described a method that controls the camera convergence and interaxial separation over time to optimally map a dynamically changing scene to the desired depth range, which improves comfort. Lang et al. [2010] proposed a method that controls and re-targets the depth of a stereoscopic scene in a nonlinear and locally adaptive fashion. The solution employs computed disparity and saliency estimates to compute a deformation of the input views so that they meet the desired disparities. To avoid undesirable distortions from disparity manipulation, Didyk et al. [2011] introduced a perceptual model of disparity, which provides a metric to evaluate perceived disparity change for stereo images. The follow-up work [Didyk et al., 2012b] studies the interplay of contrast and disparity on depth discrimination. Based on their disparity-perception model, they jointly manipulate luminance contrast and disparity to reduce depth in stereoscopic images. However, these models ignore the impact of display luminance, which is central to our work presented in Chapter 7. Didyk et al. [2012a] also proposed a depth-enhancement technique that relies on the Cornsweet illusion in the disparity domain. While all these methods show the potential of manipulating disparity to improve the perception of depth in a displayed image, we argue that manipulating disparity in VR content might affect visual feedback to egomotion and contribute to an intensified VR sickness [Jacobs et al., 2019].

Depth Enhancement by Image Content Manipulation It has been shown that certain image manipulations can enhance the apparent depth. Luft et al. [2006] proposed a technique that enhances contrast and color near depth discontinuities to improve the perceptual quality of monoscopic images. In Chapter 7, we use image content manipulation methodology to preserve the perception of depth in stereoscopic content across different luminance levels rather than to enhance it.

2.1.5 Perception in Low Luminance Conditions

In this section, we first consider works that discuss the advantages and disadvantages of dimming a display with an emphasis on the VR setting. Next, we discuss studies on the functioning of the human visual system (HVS) and color appearance in dark and bright conditions. Finally, we outline the works that manipulate image content to improve stereo vision.

Effect of Display Dimming on User Experience Several works have studied the impact of display brightness on user experience and power consumption. Schuchhardt et al. [2015] proposed an optimal dimming scheme to reduce mobile display brightness while ensuring good legibility on the screen. Erickson et al. [2020] investigated the effect of color mode on visual acuity and fatigue with VR head-mounted displays. They found that a dark background used in dark mode can reduce visual fatigue and increase visual acuity in a dim VR environment. Mantiuk et al. [2009] argued for using amber and red colors on dark displays as they induce the least

amount of disability glare or photophobia. They also found that the preferred display brightness was between 20 and 40 cd/m^2 in a dark environment. Chapiro et al. [2019b] reported that, in low luminance conditions, the judder is less visible, leading to better-perceived motion quality. These works provide solid ground for the merit of dimming VR displays. However, it is well recognized that the visual performance, including contrast and depth perception, is substantially degraded at low luminance. Although the visual system can preserve the appearance of contrast through a range of conditions [Georgeson and Sullivan, 1975], the contrast appears weaker and eventually disappears as the luminance is reduced, particularly the contrast that is close to the threshold [Kulikowski, 1976; Peli et al., 1991]. Lower luminance levels cause the pupil to dilate. This could result in a larger defocus blur in fixed-focus displays. Singh et al. [2018] found that matching the brightness of the displayed and real object on an AR display results in more accurate depth estimation when focus cues are consistent (no vergence-accommodation conflict) or when focused on the mid-point of the tested depth range. No absolute luminance levels were reported, so we cannot compare their finding to ours. According to Frisby and Mayhew [1978], the global stereopsis mechanism requires a certain level of suprathreshold contrast to detect a binocular disparity signal. Since the contrast detection thresholds are much higher at low luminance, our ability to see depth in low contrast content is greatly reduced [Livingstone and Hubel, 1994]. Chapter 7 focuses on solving this issue.

Color Appearance on Dimmed Displays Despite the ability of our visual system to maintain color perception across a very wide range of illumination (color constancy), some changes in appearance are inevitable. Kim et al. [2009] observed that at higher luminance levels perceived colorfulness and lightness increase. The opposite effect can be observed when light levels are low, in particular when the visual system transitions from cone-mediated vision (photopic) to cone- and rod-mediated vision (mesopic) [Barbur and Stockman, 2010]. Indeed, color appearance in mesopic vision (0.01 - 3 cd/m^2) can be influenced by the change in rod activity [Stabell and Stabell, 1998]. As a consequence, luminance levels alter the perception attributes such as hue, chroma, and lightness [Kwak et al., 2003; Shin et al., 2004]. Brightness and colorfulness also reduce with decreasing luminance [Fu et al., 2012]. Several models explaining the changes in color appearance at mesopic light levels have been proposed [Shin et al., 2004; Cao et al., 2008], including an extension of CIECAM02 color appearance model [Luo and Li, 2013].

Impact of Color on Depth Perception Multiple studies tried to investigate the impact of colored stimuli on perceived depth. Trościanko et al. [1991] found out that certain isoluminant color gradients (the ones occurring in nature) might be used as a monocular depth cue, but its input to stereopsis is weak. The follow-up work [Bailey et al., 2007] established that the influence of colors on depth judgments is even weaker when natural images are used as stimuli. Overall, chromatic (isoluminant) contrast has little influence on the disparity depth cues [Simmons and Kingdom, 1994]. For that reason, the vast majority of binocular vision studies rely on achromatic stimuli (stereo acuity thresholds are ten times larger for chromatic stimuli than for achromatic ones [Krauskopf and Forte, 2002]). Because chromatic contrast has little influence on depth perception, our considerations are limited to luminance.

Simulation and Compensation of Night Vision Some works tried to simulate and compensate for changes between day and night visions. Wanat and Mantiuk [2014]

proposed a luminance re-targeting method to match the appearance of different luminance levels by altering perceived contrast and modeling of hue and saturation shifts of an image. Kellnhofer et al. [2014] argued that for stereoscopic displays, such changes are not sufficient to fully simulate dark conditions. Their proposed solution involves the manipulation of binocular disparity so that a scotopic stereo content displayed on a photopic monitor is perceived as the scene was scotopic. Contrary to the mentioned studies, instead of improving the appearance of a dark screen [Wanat and Mantiuk, 2014], or simulating the scotopic vision on bright display [Kellnhofer et al., 2014], Chapter 7 focuses on improving stereo vision in low luminance conditions.

2.1.6 Masking

Masking refers to a phenomenon in which the detection or perception of one visual stimulus is affected by the presence of another stimulus. The effect of masking on visual perception can be influenced by various factors such as luminance, contrast, and spatial relationship between the target and mask. The strength of the masking effect can also depend on individual differences in visual processing, such as age, attention, and visual acuity. In this section, we will discuss two types of masking: spatial contrast masking and luminance masking.

Spatial Contrast Masking

There might be situations when two stimuli interfere with each other's perception, making it more difficult to distinguish one from the other. For it to happen, the stimuli need to have similar contrast spatial frequency and orientation. When these requirements are fulfilled, one signal can mask another. To understand better the mechanism of contrast masking, we shall analyze Figure 2.8.

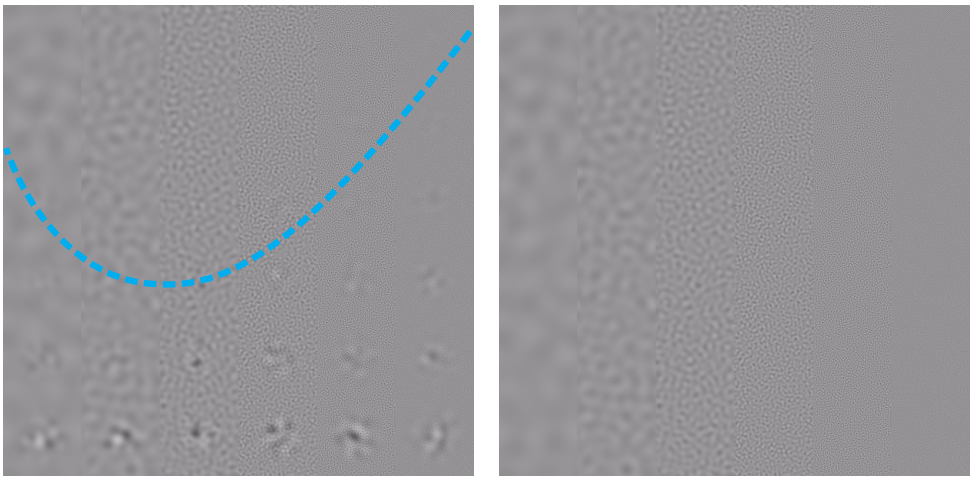


FIGURE 2.8: *An example of contrast masking. The image on the left showcases the summation of the target and masking signals. Additionally, the masking signal is provided for reference (right). When the spatial frequency of the masking signal is similar to that of the target signal, spotting the target signal in their sum is difficult. However, when the spatial frequencies of both signals differ significantly, the test signal becomes much easier to detect. A blue dashed line represents the approximated visibility threshold.*

The figure showcases the combination of the target and masking signals. Throughout the entire image, the frequency of the target signal remains constant, while the frequency of the masking signal varies along the horizontal axis (as evident in the

right image). It can be observed that when the frequencies of both signals are similar, a significantly higher amplitude of the target signal is required to distinguish it from the accumulated signal. However, when the frequency of the masking signal is lower or higher than that of the target signal, this task becomes much easier. This phenomenon is represented by the blue dashed line, which approximates the visibility threshold. It's important to note that a similar effect can be observed regarding the orientation of the stimulus (though not demonstrated in the noise example in the figure). When the orientations of both stimuli are similar, detecting the target signal becomes challenging, while the task is the easiest when the signals have perpendicular orientations.

Luminance Masking

The sensitivity of the human eye at different levels of luminance varies greatly. The same absolute luminance difference can be distinguished in low light conditions, while it is imperceptible when the base luminance is high. This phenomenon, commonly referred to as luminance masking or luminance self-masking, arises from the application of the Weber law, which asserts that the smallest noticeable absolute difference in luminance increases proportionally with the background luminance. It is natural to assume that Weber law predicts how we perceive distortions based on the surrounding luminance. To account for these differences in perception, luminance can be transformed into the logarithmic domain. However, the logarithmic transform, known as Fechner's law (an extension of Weber's law), is not an exact model of the human visual system's sensitivity to changes in light [Mantiuk et al., 2006]. Instead, the threshold vs. intensity or contrast sensitivity function is used to determine the smallest perceivable difference in luminance across the range of luminance values. This information is used to create a mapping function that approximates perceptually uniform units from physical luminance values. Nowadays, luminance masking is well understood and can be easily modeled [Mantiuk et al., 2004a; Mantiuk et al., 2004b; Mantiuk et al., 2011a; Aydin et al., 2008a; Miller et al., 2013].

2.2 State-of-the-Art Metrics

Quality metrics are essential in evaluating the accuracy and efficiency of digital media processing algorithms. In this section, we will discuss the state-of-the-art works related to quality metrics in multiple contexts. We will start with the image quality metrics (Section 2.2.1), which often inspire the design of mesh quality metrics (Section 2.2.2). Finally, in Section 2.2.3, we will discuss existing motion quality metrics that investigate the temporal domain of the content and characteristics of the displaying hardware.

2.2.1 Image Quality and Visibility Metrics

Image metrics can be divided into *quality* and *visibility* metrics, both addressing different applications. *Image quality metrics (IQMs)* predict a single global quality score for the entire image. These metrics usually are trained and evaluated on mean opinion scores (MOS) [Sheikh et al., 2006; Ponomarenko et al., 2015a] that are obtained in user experiments for each distorted image. In contrast, *visibility metrics* [Daly, 1992b; Aydin et al., 2008b; Mantiuk et al., 2011b] predict the probability that a human observer will detect differences between a pair of images. They provide localized information in the form of visibility maps, in which each value represents

a probability of detection. Visibility metrics tend to be more accurate for small and barely noticeable distortions but are unable to assess the magnitude of distortion. Visibility metrics are often more relevant for graphics applications whose goal is to maximize performance without introducing any visible artifacts.

The initial chapters of this work focus on visibility metrics that are suitable for computer graphics applications. In this section, we provide an overview of previous quality and visibility metrics with a focus on the latter ones. Since we use machine learning techniques to derive our metric, we also discuss relevant literature in this area. Concluding this segment, we touch upon attention models, which could potentially enhance metric accuracy in certain applications.

Quality Metrics

The vast majority of IQMs are *full reference* (FR) techniques that take as input both reference and distorted images and compute local differences, which are then pooled across the entire image into a single, global quality score. The simplest approach to compute such local differences is pixel-wise absolute difference (ABS), or the Euclidean distance (ΔE) between RGB components. The latter is employed in the popular Root Mean Square Error (RMSE) and Peak Signal-to-Noise Ratio (PSNR) metrics. A better approximation to the perceived differences is achieved when pixel RGB values are converted to a perceptually uniform color space and a color difference formula, such as CIE ΔE 2000 (CIEDE2000), is used.

More advanced quality measures such as the Structural Similarity Index Metric (SSIM) [Wang and Bovik, 2006, Ch. 3.2] account for spatial information by computing differences in the local mean intensity and contrast, as well as cross-correlation between pixel values. The Visual Saliency-Induced Index (VSI) employs a similar framework but contributes to the local difference map's four components: the visual saliency, the gradient magnitude, and two chrominance channels [Zhang et al., 2014]. The Feature Similarity Index (FSIM) also employs the gradient magnitude, this time complemented by the phase congruency, to derive the local difference map [Zhang et al., 2011]. VSI employs a saliency map, and FSIM the phase congruency map, as a weighting function used for pooling the final score. While VSI and FSIM produce local difference maps at intermediate processing stages, their utility as local visibility predictors has not been tested. Although global IQMs, such as SSIM, VSI, and FSIM were not intended to predict visibility, in Chapter 3, we demonstrate that they can be trained as such.

For a complete overview of IQMs, we refer the reader to numerous surveys [Lin and Kuo, 2011; Chandler, 2013], which also provide information on over 20 image quality databases with MOS data, including the popular LIVE [Sheikh et al., 2006] and TID2013 [Ponomarenko et al., 2015a] datasets. The distortion types covered in those datasets correspond to the most prominent applications of IQMs and include various image compression and transmission artifacts, as well as different types of noise, blur, and ringing.

Visibility Metrics

Visibility metrics address the challenging problem of predicting the visibility of distortions for each pixel location. Since it is more difficult to collect sufficient data for training such visibility metrics, they often rely on the low-level models of the visual system. The models help to constrain a space of possible solutions and reduce the number of parameters that need to be trained.

Early works on visibility predictors often focused on modeling spatial contrast sensitivity. For example, the sCIELab [Zhang and Wandell, 1997] metric prefiltered CIELab encoded pixels with a spatio-chromatic contrast sensitivity function prior to computing the visibility map. This simple approach has limited success in predicting distortions in complex images, as it ignores important aspects of supra-threshold perception, such as contrast constancy or contrast masking. Those issues are addressed by a family of more complex metrics inspired by the models of low-level vision. Some examples of those are the Visual Discrimination Model (VDM) [Lubin, 1995], the Visible Differences Predictor (VDP) [Daly, 1993], and HDR-VDP [Mantiuk et al., 2011b]. Those metrics consider luminance adaptation, contrast sensitivity, contrast masking, and frequency-selective visual channels [Chandler, 2013]. However, their predictions are less accurate for complex images [Čadík et al., 2012].

A different approach was taken by metrics based on feature maps, such as Butteraugli, which is the core part of Google's perceptually guided JPEG encoder project "Guetzli" [Alakuijala et al., 2017]. Butteraugli transforms an input image pair into a set of feature maps, such as an edge detection map and a low-frequency map, which are then combined to form the final difference map prediction. The maximum value in the difference map is taken as the score of JPEG compression quality. As the metric was intended for JPEG artifacts, it is not clear how this metric performs for computer graphics applications that are notoriously difficult for other visibility metrics [Čadík et al., 2012].

In Chapter 3, we support the observation that the key problem that hinders the development of better quality metrics is limited training data, which must be locally annotated and represent a great variety of distortions with sub-threshold, near-threshold, and supra-threshold magnitudes [Chandler, 2013]. In an attempt to fill this gap, Alam et al. [2014] measured local discrimination thresholds for over 3000 patches but in only 30 images. The measurement of discrimination thresholds is a very tedious task that is not well suited for collecting large datasets. Therefore, in this work, we use a more efficient experimental procedure of marking visible distortions [Herzog et al., 2012; Čadík et al., 2012; Čadík et al., 2013].

We extend the range of collected data not only by considering a much larger set of images and distortions but also by measuring different levels of these distortions, which we found to be essential for successful training. We propose an efficient experimental setup to reduce the within-subject variance in such systematic distortion marking, which further improves the consistency of our training data. We capitalize on the availability of such massive training data to improve the performance of existing visibility metrics such as sCIELab, HDR-VDP, and Butteraugli. Also, we attempt to further improve the accuracy of local visibility predictions with a new CNN-based metric. We expect that through extensive training, relevant characteristics of the human visual system can be learned, including higher-level effects related to the image content, its complexity, and distortion type.

CNN-Based Quality Metrics

Although the utility of CNN-based methods has not been demonstrated in the context of visibility metrics so far, several solutions for IQMs have been proposed. We discuss here selected IQMs that show some similarities to the method proposed in Chapter 3 regarding the architecture or training methodology. Since, in all cases, huge volumes of training data are required, those metrics are trained on small patches extracted from image pairs [Bianco et al., 2016; Kang et al., 2014; Bosse et al., 2016b; Bosse et al., 2016c]. The key problem is that all patches derived from an image share the same MOS value.

Although this seems to be a reasonable assumption for homogeneous distortions, such as compression or noise, even in such cases, the artifact visibility varies across the image due to complex HVS effects such as contrast masking [Chandler, 2013; Mantiuk et al., 2011b]. Furthermore, when dealing with spatially-varying distortions, which are common for computer graphics applications, using a single value per image is not an option. In Chapter 3, we avoid this problem in our training by using dense visibility maps from the marking experiment.

Inspired by observations that low-level features should be able to capture natural scene statistics [Wang and Bovik, 2006], Kang et al. [2014] designed a shallow CNN architecture consisting of five layers and including only one convolutional layer with 50 filter kernels. Although the complexity of the architecture is similar to the traditional quality metrics, the key difference is that in the case of CNN-based metrics, the kernels are learned. Despite the good performance of shallow architectures, it has been demonstrated that a much deeper 12-layer CNN can achieve further improvements [Bosse et al., 2018; Amirshahi et al., 2016]. This suggests that higher-level features encoding locally invariant information important for object recognition may contribute to the image quality evaluation.

A similar architecture was applied in the context of FR-IQMs [Bosse et al., 2018]. The key idea was to duplicate the previous architecture for both the reference and distorted images and then submit the resulting feature vectors to a 2-layer fully connected network to learn a regression function that estimates the aggregated MOS rating. To address the spatial variance of relative image quality, a second branch is integrated into the regression module to learn the importance of patches, i.e., their relative weight in the aggregated MOS rating. This is an attempt to correct the assumption of the same MOS rating for all patches in a distorted image, which was made for the training data.

Sim et al. [2020] presented a novel approach based on a neural network that does not require any additional training, as it uses the convolutional layers of a pre-existing model. The technique combines local quality scores within a deep similarity map using mean and deviation (MaD) pooling, which employs two essential descriptive statistics: the weighted mean, which reflects the impact of visual saliency on quality, and the standard deviation, which reflects the impact of distortion distribution within the image.

As transformer architecture became successful in multiple visual tasks, Cheon et al. [2021] proposed an Image Quality Transformer (IQT) that applies a transformer architecture to a perceptual full-reference image quality assessment task. Perceptual representation is crucial in this context, and the IQT extracts feature representations using a convolutional neural network backbone. The transformer encoder and decoder are then used to compare the reference and distorted images with learnable quality and position embeddings.

While it had been a common approach to predict image quality by explicitly comparing features and measuring the severity of image distortion based on how far the corresponding feature is from the reference image space, [Liao et al., 2022] proposes a statistical distribution approach to model quality degradation in perceptual space. The proposed method measures quality based on the Wasserstein distance in the deep feature domain, specifically the 1D Wasserstein distance at each stage of the pre-trained VGG network, which is then used to determine the final quality score. The proposed DeepWSD approach offers better interpretability of quality contamination caused by various types of distortions and has advanced quality prediction capability.

In Chapter 3, we propose a visibility metric that is also based on CNN and greatly benefits from our adequate training data. Moreover, it features a novel CNN

architecture that significantly departs from architectures in image quality evaluation considered at the time of publication. In our loss function, we explicitly account for the stochastic nature of training data due to differences between subjects as well as the combined effect of distortion search and actual detection in complex images.

Attention Models

Depending on application of a given image metric, one might consider enhancing it by applying various attention models. Such models are computational frameworks that simulate how attention operates in human cognition and aim to mimic the selective processing of information that occurs in the human brain due to limitation of its capability to process all stimuli at the same time. Attention can be categorized into two distinct functions: bottom-up attention, in which attentional guidance is primarily driven by the salient features of the stimulus, such as color or contrast, and top-down attention, referring to guidance of attention based on internal goals, expectations, or task demands.

One of the first computational saliency-based model which can be attributed to the first trend was proposed by Itti et al. [1998]. In addition to the bottom-up processing, focusing on low-level features of an image such as color, intensity, and orientation of the features, it considered also the center-surround mechanism and feature contrast to simulate the involuntary, stimulus-driven aspects of human attention. This work has inspired multiple follow-up works which, for example, further expanded it to spatio temporal domain [Le Meur et al., 2006] or by incorporating signals specific to parvocellular and magnocellular cells of the retina, enabled a more nuanced representation of visual attention, thus improving accuracy [Marat et al., 2009]. Borji and Itti [2013] provide more details on this topic in their recent survey. As mentioned earlier in the section, the attention mechanism has also served as a source of inspiration in the field of image quality assessment. It has been effectively employed in the VSI metric, where visual saliency acts as a pooling function to determine the final quality score.

As Chapter 3 describes attempts on desinging a highly-conservative general-purpose visibility metric, which predicts visibility with an assumption that the observer knows where the distortion occurs, we do not integrate attention models into it.

2.2.2 Mesh Quality Metrics

The image metrics described in the previous section operate in "pixel space", i.e., the objects of comparison are the final rendered images, and they do not address geometric distortions on underlying meshes, if present. In contrast to this, it is often desirable to evaluate the quality of 3D meshes at an earlier stage of the graphics pipeline, while staying agnostic to illumination and texture long before the final image is rendered. 3D mesh quality metrics tackle this problem. Unfortunately, despite the significant attention and exploration of image metrics by a sizable community, mesh quality metrics have not received the same level of attention. This section outlines several of the most widely used mesh quality metrics.

A popular geometry metric is Metro [Cignoni et al., 1998], which uses a Hausdorff distance as a base component. Metro is mathematically motivated and does not aim to reproduce perception. In contrast, metrics like MSDM [Lavoué et al., 2006] are inspired by perceptually-motivated image metrics like SSIM and use mean curvature as input—informing on the smooth vs. bumpy profile of the studied object. Local

distortions are measured as a difference of Gaussian-weighted statistics over a local spherical neighborhood, with the final score being computed through Minkowski pooling. MSDM2 [Lavoué, 2011] improves upon the previous version by projecting the curvature value between two objects and using a multiscale approach, allowing users to compare meshes with different topologies. More advanced perceptual models inspired the authors of DAME [Váša and Rus, 2012], which takes visual masking into account. Following the intuition that distortions introduced to bumpy areas are less visible than ones added to smooth surfaces, the authors calculate dihedral angles and weigh them accordingly. This metric is also significantly faster than MSDM, making it more suitable for computationally intensive tasks like optimization. Another fast roughness-based method is FMPD [Wang et al., 2012], which computes the roughness of each vertex on the test and reference meshes using local Gaussian curvature, modulated by a power function to model visual masking. The final score is computed by taking the difference between normalized surface integrals of each mesh. This method is unsuitable for low-frequency distortions, as their impact can be underestimated.

TPDMPW [Feng et al., 2018] proposes a novel spatial pooling method to assess mesh quality, based on applying a percentile weighting strategy to existing metrics to emphasize local regions with severe distortions. Nouri et al. [2016] propose a multiscale saliency map as a basis for local statistics on 3D meshes, computing differences in mean local roughness to model visual masking. Nader et al. [2015] introduces a just-noticeable-difference-based 3D geometry distortion metric developed for flat-shaded geometries. The authors employ a sensitivity paradigm, which models perceptually relevant parameters like distance and masking.

2.2.3 Motion Quality Metrics

Motion quality metrics aim to predict the visibility of motion artifacts given content and display information as input. Such models can then be used for various motion-adaptive rendering applications. Primarily, there are two main approaches to model quality: black-box metrics and white-box metrics. Black-box metrics [Chapiro et al., 2019a; Debattista et al., 2018] explain the data by fitting an arbitrary function and typically perform well within the domain of the dataset but are unable to extrapolate beyond it. White-box metrics, on the other hand, rely on psychophysical models and are better at extrapolating the predictions. We focus on white-box metrics as we aim to build a metric that generalizes to a wide range of display technologies and content.

Denes et al. [2020] proposed a white-box metric (called MARRR) that models motion quality as a weighted sum of the quality of individual motion artifacts. The metric makes a simplifying assumption that motion artifacts are independent of each other, which fails for content-adaptive scenarios where blur and spatio-temporal aliasing from rendering are often inter-dependent. Hoffman et al. [2011] analyses the frequency spectra of a moving line to predict the visibility of motion artifacts, however, it is unknown if such predictions will extend well to supra-threshold appearance. Both metrics ignore the impact of content on motion quality.

Yang et al. [2019] proposed a metric for adaptive control of the VRS, called Nvidia Adaptive Shading (NAS), which considers both the screen content and the velocity. The metric analyses the screen content in the frequency domain and uses that information together with the motion vectors to classify each VRS tile into full, half, and quarter-resolution shading. The objective of their metric is to use the lowest possible shading rate that results in visually lossless quality. Unlike our method introduced in Chapter 6, which keeps the computational cost the same across the frames, their

method will result in varying rendering costs across the frames. Therefore, their method is more suitable for applications in which the rendering budget is unconstrained, but it is desirable to keep it low, for example, to reduce power consumption. A more detailed discussion of their method can be found in Section 6.5. Perceptual video quality metrics such as [Mantiuk et al., 2021a] are also designed to be content and motion-aware, but their complexity precludes real-time rendering.

Furthermore, most of the existing models only consider a small subset of motion artifacts. One artifact that is rarely considered is temporal and spatial aliasing due to rendering (Table 2.1). Motion artifacts are often inseparable, and studying them in isolation does not translate well to practical applications. Aliasing due to rendering is also particularly important for our application as it is one of the most common artifacts resulting from the shading rate reduction. The summary of the discussed metrics can be found in Table 2.2. Our proposed metric accounts for all the important factors described in Section 2.1.3 and outperforms the above models, as demonstrated in Section 6.3.5.

TABLE 2.2: *Models of motion quality and the parameters they account for. Our solution described in Chapter 6 has been marked in bold.*

Motion Quality Metrics	Refresh Rate	Resolution	Persistence	Velocity	Luminance	Content-aware
[Hoffman et al., 2011]	Yes	No	Yes	Yes	No	No
[Debattista et al., 2018]	Yes	Yes	No	No	No	No
NAS [Yang et al., 2019]	No	Yes	No	Yes	Yes	Yes
[Chapiro et al., 2019a]	Yes	No	No	Yes	Yes	No
MARRR [Denes et al., 2020]	Yes	Yes	Yes	Yes	No	No
CaMoJAB [Jindal et al., 2021]	Yes	Yes	Yes	Yes	Yes	Yes

2.3 Technical Aspects and Tools

In this section, we will delve into the technical aspects and tools used in this dissertation. Specifically, we will discuss two critical techniques: perceptual scaling (Section 2.3.1) and Variable Rate Shading (Section 2.3.2).

2.3.1 Perceptual Scaling

To standardize the data from a subjective study and make it more practical, one may scale it into units of *just objectionable difference (JOD)* [Perez-Ortiz et al., 2019], closely related to the psychophysical concept of a *just noticeable difference*. An advantage of using a JOD scale is that it provides a universal perceptual scale of quality directly related to the probability that a distorted mesh will be chosen over another in an experiment, as shown in Figure 2.9. We expect this type of perceptually-meaningful scale to be more interpretable and to lend itself to setting useful perceptual thresholds in applications, as this type of scaling is already popular for image and video difference metrics [Mantiuk et al., 2011a; Mantiuk et al., 2021b]. We use such scaling in Chapters 5 and 6 to standardize the subjective data.

2.3.2 Variable Rate Shading

VRS [Nvidia, 2018] or coarse pixel shading [Vaidyanathan et al., 2014] are recent technologies, available on the latest GPUs. VRS decouples shading and visibility calculations to provide flexible control over shading. By rasterizing at full resolution

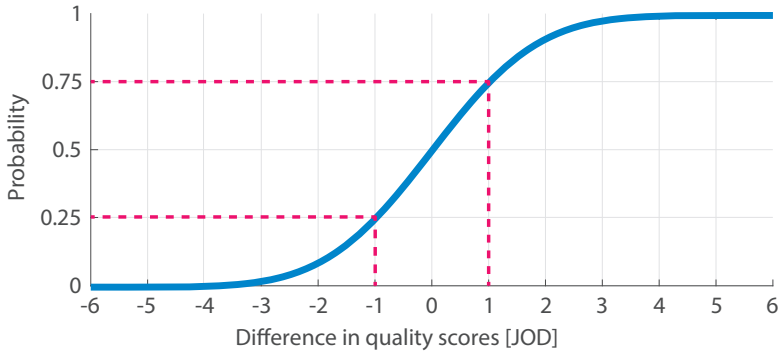


FIGURE 2.9: *This plot shows how pairwise comparison preferences are represented in* just objectionable difference *(JOD) scores. For example, if a distorted mesh* $\boldsymbol{A}$ *is 1 JOD away from the reference, this represents a 50% decrease in the likelihood of selection, i.e., the reference will be chosen over* $\boldsymbol{A}$ *75% of the time. More detail can be seen in Section 3.9 of the work of Mantiuk et al. [2021b].*

but shading at a lower resolution, the technique reduces pixel shading load while preserving edges and the visibility of objects. Unlike their predecessors, Multi-Resolution Shading and Lens-Matched Shading [Kraemer, 2018], which optimize a shading workload to match VR optics, VRS allows for much more granular control, where every 16×16 tile of pixels can have a different shading rate. The recent specification offers seven different shading rates ($1 \times 1, 1 \times 2, ..., 4 \times 4$), including non-uniform shading resolution along horizontal and vertical dimensions. Since VRS only executes the fragment shader a few times per multiple raster pixels and broadcasts the same shaded value to the neighboring pixels, the artifacts generated by VRS on the surface of each object are the same as upsampling with a box filter. The visibility of these artifacts depends on several factors such as luminance, refresh rate, velocity, and content, as can be seen in Chapter 6, where we use VRS to control the resolution of our stimuli.

Chapter 3

General Purpose Metric Design

3.1 Introduction

A large number of applications in graphics and imaging can benefit from knowing whether introduced changes in images are visible to the human eye or not. Existing visibility metrics provide such predictions [Mantiuk et al., 2011b; Alakuijala et al., 2017], but achieve only moderate success. They work well for simple stimuli, but their performance is worse for complex images. They can predict low-level visual phenomena, such as luminance and contrast masking, but they do not account for higher-level effects due to image content. We show that such higher-level effects have a significant role in detecting visible distortions.

To create a robust predictor of visible distortions, we collect a large dataset, named *LocVis*, with locally marked distortions (Section 3.2 and 3.3). The dataset includes 557 image pairs, 296 of which were manually marked by 15 to 20 observers, and the remaining 261 pairs were generated from existing TID2013 dataset. In comparison, the next largest dataset contains just 37 marked images [Čadík et al., 2012]. Moreover, our dataset contains more extensive variations in both type and magnitude of image artifacts. We model collected data (Section 3.4) and use it to calibrate popular visible difference metrics in order to improve their performance (Section 3.5). Then, we use the dataset to train a novel metric based on a CNN architecture (Section 3.6), which improves the prediction accuracy over existing metrics by a substantial factor (Section 3.7).

We demonstrate that the new CNN metric can not only predict subjective data but enables many relevant applications (Section 3.8). In the first example, we show that the CNN metric can reduce the size of JPEG images by about 60%, while achieving visually lossless compression (Section 3.8.1). In the second application, we demonstrate that the metric can determine the maximum downsampling factor for which a single-image super-resolution algorithm can reconstruct a visually equivalent image (Section 3.8.2). The third example demonstrates the utilization of a CNN-based metric to seamlessly embed an imperceptible watermark into an image (Section 3.8.3).

As the initial version of the metric can predict visibility for fixed viewing conditions, in Section 3.9 we present a way to extend proposed metric so it can account for varying viewing distance and display luminance. We end this chapter with a discussion on the limitations of the proposed metric (Section 3.10) and conclusions (Section 3.11).

The code and the LocVis dataset can be found at:
`http://visibility-metrics.mpi-inf.mpg.de/`.

TABLE 3.1: *Dataset details.*

Subset name	Scenes	Images	Dist. levels	Level generation method	Res. [px]	Source
MIXED	20	59	2-3	blending	800×600	custom software, photographs [Čadík et al., 2012]
PERCEPTIONPATTERNS	12	34	1,3	blending	800×800	MATLAB [Čadík et al., 2013]
ALIASING	14	22	1-3	varying sample number	800×600	Unity, CryEngine [Piórkowski et al., 2017]
PETERPANNING	10	10	1	n/a	800×600	Unity, CryEngine [Piórkowski et al., 2017]
SHADOWACNE	9	9	1	n/a	800×600	Unity, CryEngine [Piórkowski et al., 2017]
DOWNSAMPLING	9	27	3	varying shadow map resolution	800×600	Custom OpenGL app [Piórkowski et al., 2017]
ZFIGHTING	10	10	1	n/a	800×600	Unity, CryEngine [Piórkowski et al., 2017]
COMPRESSION	25	71	2-3	varying bit-rates	512×512	crops from photographs
DEGHOSTING	12	12	1	n/a	900×900	photographs [Karađuzović-Hadžiabdić et al., 2017]
IBR	18	36	1,3	varying distance between key frames	960×720	custom software [Adhikarla et al., 2017]
CGIBR	6	6	1	n/a	960×720	custom software [Adhikarla et al., 2017]
TID2013	25	261	n/a	n/a	512×384	Kodak image dataset [Ponomarenko et al., 2015a]

3.2 Dataset of Visible Distortions

One of the primary objectives of this chapter is to gather data regarding the visibility of distortions in each image location. This enables the differentiation between distortions that fall below the visibility threshold and cannot be detected and those that are clearly visible. Typically, visibility thresholds are determined through threshold experiments utilizing constant stimuli, adjustment, or adaptive methods. However, these methods tend to measure only one image location at a time, resulting in highly inefficient procedures. For instance, the largest dataset obtained using such methods [Alam et al., 2014] consists of only 30 images, each with dimensions of 216×216 pixels, and it required numerous hours of experimentation to collect.

To overcome these limitations, we have refined the procedure outlined by Čadík et al. [2012] to create the most extensive dataset of locally visible distortions (shortly LocVis dataset). Our dataset comprises 557 images with 170 unique scenes, many of which are generated with up to 3 levels of distortion, such as varying image compression quality settings. The scenes were carefully selected to encompass a wide range of common and specialized computer graphic artifacts, including noise, image compression, shadow acne, peter-panning, warping artifacts from image-based rendering methods, and deghosting resulting from HDR merging. This diverse collection of distortions poses a challenge for existing visibility metrics. To facilitate easy access to our data, we have organized the images into subsets containing similar artifact types. A terse summary of the dataset can be found in Table 3.1, while a description of each subset can be found in the following paragraphs.

MIXED This subset expands upon the LOCCG dataset introduced by Čadík et al. [2012]. In this extension, we create images with multiple distortion levels by blending or extrapolating the variations between the distorted and reference images. The dataset comprises 59 images, encompassing a range of distortions such as high-frequency and structured noise, virtual point light (VPL) clamping, light leaking artifacts, local brightness changes, and tone mapping artifacts. An example of VPL-induced artifacts can be observed in Figure 3.1.



FIGURE 3.1: *Example of distorted shadows and local brightness changes, common artifact in VPL method.*

PERCEPTIONPATTERNS The Contrast-Luminance-Frequency-Masking (CLFM) dataset comprises 34 images obtained from work by Čadík et al. [2013], featuring artificial patterns specifically created to demonstrate prominent perceptual phenomena such as luminance masking, contrast masking, and contrast sensitivity. Unlike other sets in our collection, this particular dataset includes abstract patterns like blobs or stripes with varying contrast values. We prepared three levels of distortion for these scenes by linearly blending reference and distorted images. Figure 3.2 provides an example of this subset.

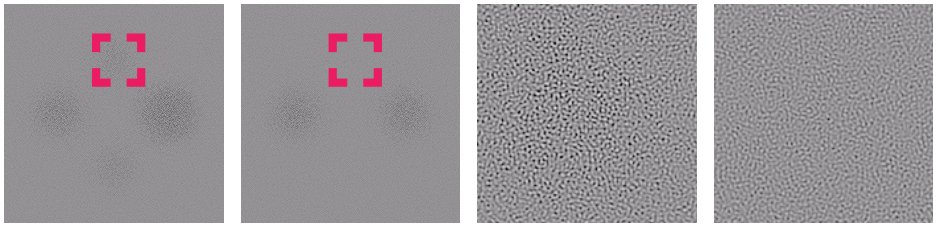


FIGURE 3.2: *Example of perception patterns dataset.*

ALIASING Aliasing occurs when the sampling frequency of a signal is insufficient to accurately reproduce high-frequency details, as per the Nyquist criterion. In the context of images, aliasing manifests as jagged profiles, inaccurately rendered details, and stair-step artifacts along the edges. To address this type of artifact, we selected scenes from the work of Piórkowski et al. [2017] and rendered them using popular game engines such as Unreal Engine 4 and Unity. We created up to three levels of distortion by varying the sample numbers of the multi-sampling anti-aliasing method. Figure 3.3 showcases an example of the aliasing artifact.



FIGURE 3.3: *Example of the aliasing artifact.*

SHADOWACNE Shadow acne occurs due to a phenomenon known as self-shadowing and is a result of the discrete nature and limited resolution of the shadow map used in real-time rendering. The shadow map discretizes the scene into a grid of texels, and each texel represents a certain area of the scene. When determining whether a point on a surface is in shadow or illuminated, the renderer checks the corresponding texel in the shadow map. However, due to the limited resolution of the shadow map, there can be imprecision in representing the transition between shadowed and illuminated regions. This can lead to the supra-threshold artifact in the form of a striped Moiré pattern. The images used in this context are sourced from work by Piórkowski et al. [2017]. An example showcasing this kind of artifact can be seen in Figure 3.4.

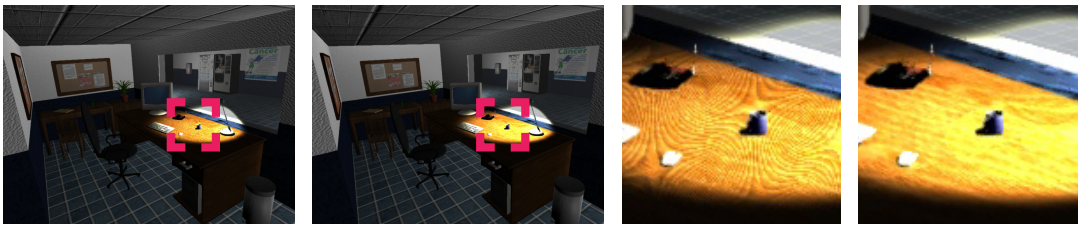


FIGURE 3.4: *Example of Moire pattern known as shadow acne artifact.*

PETERPANNING Peter Panning is a distinct supra-threshold distortion, which manifests as the appearance of objects with missing or detached shadows, creating an illusion of floating above the surface. Similar to shadow acne, the visibility of Peter Panning is influenced by the precision limitations of the depth buffer and is often a consequence of attempting to address the former issue. Shadow acne can be mitigated by applying a depth offset technique, which involves incrementing the pixel's depth based on its position in the light space. However, when this depth offset becomes excessively large, errors in the depth test occur, leading to the occurrence of Peter Panning. An example illustrating the Peter Panning artifact can be observed in Figure 3.5.



FIGURE 3.5: *Example of the Peter Panning artifact - typical 'detached' shadows.*

DOWNSAMPLING Using the smallest-possible shadow maps is a common practice for maintaining good performance in real-time rendering. However, the undersampled shadow map may result in aliased edges of the rendered shadows. This supra-threshold artifact can be challenging to notice due to its localized nature. Figure 3.6 illustrates an example of an artifact caused by low shadow map resolution.



FIGURE 3.6: *Example of jagged shadows' edges as a typical artifact that appears when the shadow map resolution is too low.*

ZFIGHTING Z-fighting (or stitching) is a 3D rendering effect that happens where two or more primitives have close values in the z-buffer. In such a scenario rendering order of these primitives cannot be determined, and they are being displayed inconsistently in front or behind each other, causing artifacts in both spatial and temporal domains. Since it is usual in game engines to deal with very complex scenes with many objects, it is quite common to experience this kind of artifact. This subset contains 10 images from work by Piórkowski et al. [2017], which were rendered in Unity and CryEngine. Figure 3.7 shows an example of a Z-fighting artifact.



FIGURE 3.7: *Example of Z-fighting artifact caused by small precision of the depth buffer.*

COMPRESSION The compression dataset consists of 71 images and contains distortions due to experimental low-complexity image compression, operating at several bit rates. Compression artifacts are the most common ones in computer graphics. Low-quality settings of compression result in very well-known blockiness or mosaic artifacts (Figure 3.8), which greatly impacts overall image quality. The distortion appears globally in the whole image, and its visibility depends on the local image content. Usually, it is well-seen on gradients and can be easily masked by some specific frequencies. This set is an important source of near-threshold distortions.

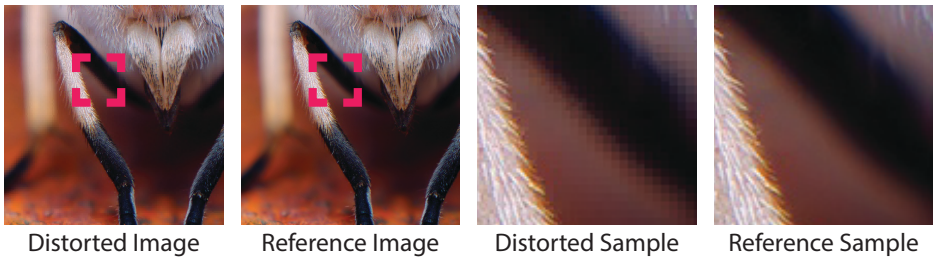


FIGURE 3.8: *Example of blockiness effect - compression artifact.*

DEGHOSTING Merging multiple exposures is a widely employed technique for generating high dynamic range (HDR) images. However, when capturing scenes with dynamic elements, non-static objects can introduce ghosting artifacts. Deghosting algorithms aim to address this issue by replacing detected motion pixels with pixels from either a single exposure or multiple exposures. However, these methods suffer from certain limitations, such as reducing the dynamic range of moving objects and causing local color or brightness deviations (see Figure 3.9). Additionally, the deghosting process can introduce other common artifacts, including motion artifacts and noise. These artifacts can manifest as supra-threshold or near-threshold distortions. The images used in this context are obtained from the work conducted by Karađuzović-Hadžiabdić et al. [2017].

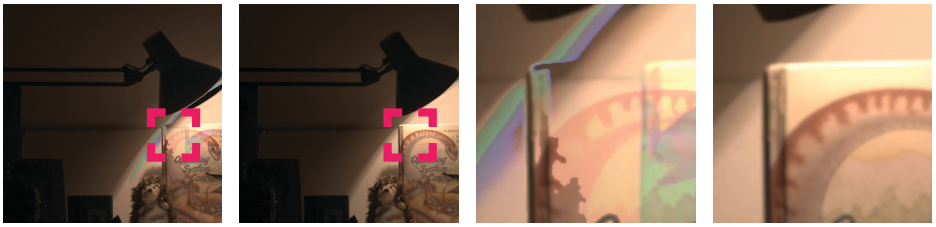


FIGURE 3.9: *Example shows local color and brightness deviations.*

IBR and CGIBR These datasets encompass distortions commonly associated with image-based rendering algorithms. Optical flow warping methods often lead to object deformation, slight ghosting, and discontinuities on the edges (Figure 3.10, top). Linear interpolation results in ghosting artifacts that appear in the image as objects with double edges and semi-transparent areas between those edges (Figure 3.10, bottom). Images generated using the nearest neighbor (NN) method do not contain any artifacts but exhibit minor shifts relative to the reference image. These shifts are nearly imperceptible to the human eye, even when the test image is compared directly with the reference. Introducing these subtle shifts was intentional to ensure the robustness of our metric against slightly misaligned images. All the images in these datasets were obtained through dense light-field camera acquisition, followed by image reduction and subsequent reconstruction using either optical flow, linear interpolation, or nearest neighbor policies. The NN images and CGIBR dataset have a single distortion level, while the optical flow images have three distortion levels. All scenes in this subset originate from work by Adhikarla et al. [2017].

TID2013 In addition to 296 newly marked images, we added 261 images from the TID2013 image quality dataset [Ponomarenko et al., 2015a], for which we could automatically generate marking. We selected from that dataset a subset of images that did not contain noticeable differences and assigned them marking maps with a value of all pixels set to zero (no user markings). Then we selected another subset with well-noticeable distortions and set corresponding marking maps to one (distortions visible in the entire image). To ensure that both subsets were correctly selected, we compared the four least severe distortion levels with the reference images in an additional pairwise comparison experiment (comparisons missing in the original dataset) and scaled the original (per-observer) pairwise data together with additional measurements using methods described in work by Perez-Ortiz and Mantiuk [2017] and assuming Thurstone Case V observer model. Then, we selected for the first

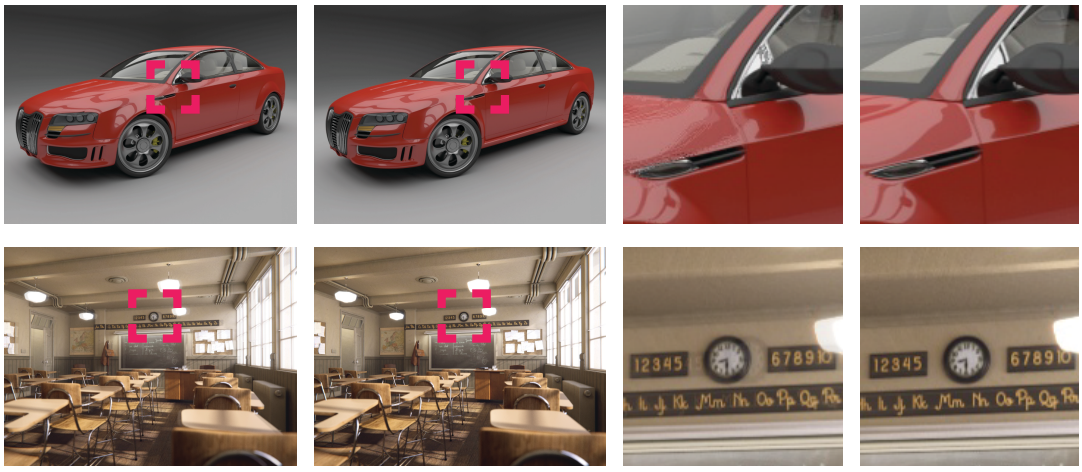


FIGURE 3.10: *Double edges and discontinuities caused by optical flow warping method (top) and ghosting artifacts caused by linear interpolation (bottom).*

subset the images with a score of less than 0.2 just-objectionable-difference (JOD) to the reference, and for the second subset, the images with a difference larger than 3 JODs. We also excluded the distortion types that affected only small image regions, such as JPEG transmission errors, and left the distortions that affected all pixels. Two selected examples from this dataset can be seen in Figure 3.11.

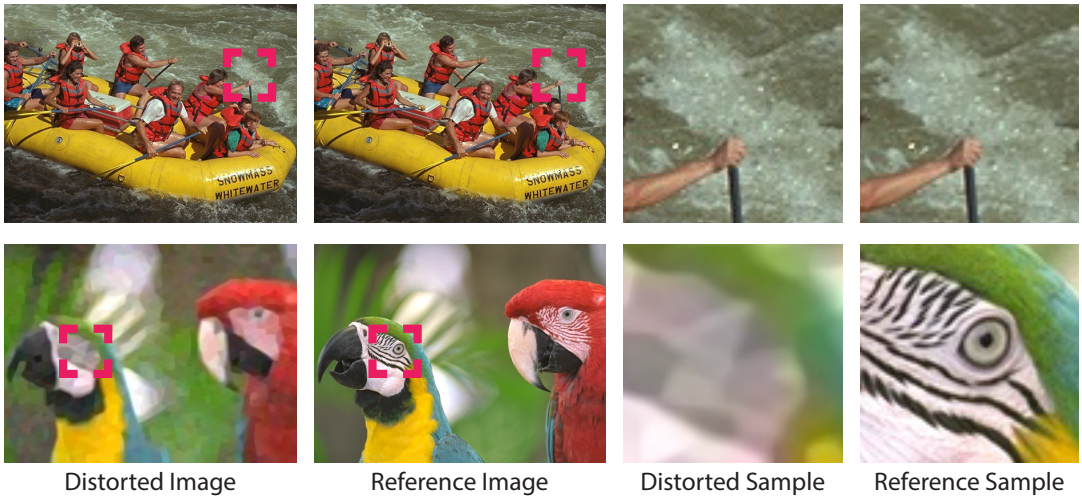


FIGURE 3.11: *Examples of imperceptible (top) and clearly-visible (bottom) artifacts from* ID2013 *subset.*

3.3 Experimental Procedure and Apparatus

In this section, we present our experimental procedure for local marking of visible distortions.

Comparison Method The visibility of image differences can be measured using different presentation methods, such as flickering between distorted and reference images, the same with a short blank screen in between, a side-by-side presentation, or no-reference presentation [Čadík et al., 2012]. Different presentation methods

will result in different sensitivity to distortions. Observers are extremely sensitive to differences in flicker presentation, resulting in overly conservative estimates of visible differences for most applications in which a reference image is rarely presented or available. For that reason, we opted for a side-by-side presentation, which is also more relevant for many graphics applications.

Experiment Software For the purpose of collecting training data, we designed a web application for marking visible distortions. To increase the comfort and accuracy of marking, we provided the ability to change brush size, erase, clear all marking, and a "lazy mouse" modifier. The "lazy mouse" function makes the brush follow the cursor slowly and lets the observer paint smooth strokes without using advanced devices while significantly increasing marking precision. Figure 3.12 depicts the application layout.

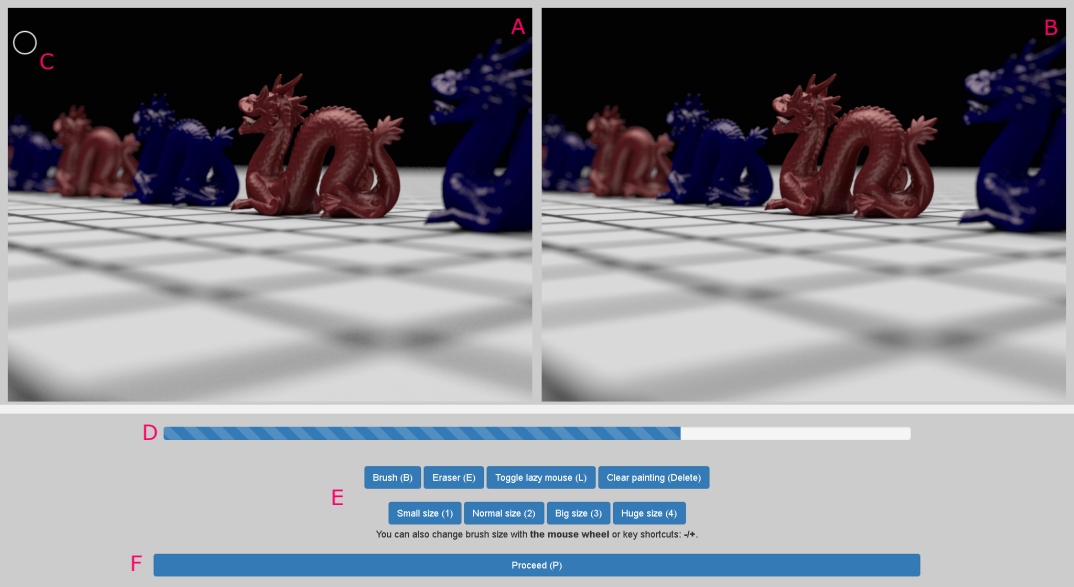


FIGURE 3.12: *Layout of the custom application for marking visible distortions: A) distorted and B) reference images, C) brush cursor, D) progress bar, E) setting buttons, and F) proceed button.*

Multiple Levels of Distortion Magnitude To increase the efficiency of collecting data and consistency, the images were presented with gradually increasing distortion magnitude. Up to three distortion levels were generated, depending on the distortion type. The marking map was copied from the previous distortion level so that only newly visible distortions had to be marked. Moving back to the previous distortion level was not allowed. Figure 3.13 shows a sample scene with three distortion levels and the corresponding observer markings.

Viewing Conditions The experimental room had dimmed lights, and the monitor was positioned to minimize screen reflections. The observers sat 60 cm from a 23″, 1920×1200 resolution Acer GD235HZ display, resulting in the angular resolution of 40 pixels per visual degree. The measured peak luminance of the display was 110 cd/m^2 and the black level was 0.35 cd/m^2. For COMPRESSION and DEGHOSTING sets, the distance was changed to the one corresponding to 60 pixels per visual degree to reduce the visibility of distortions.

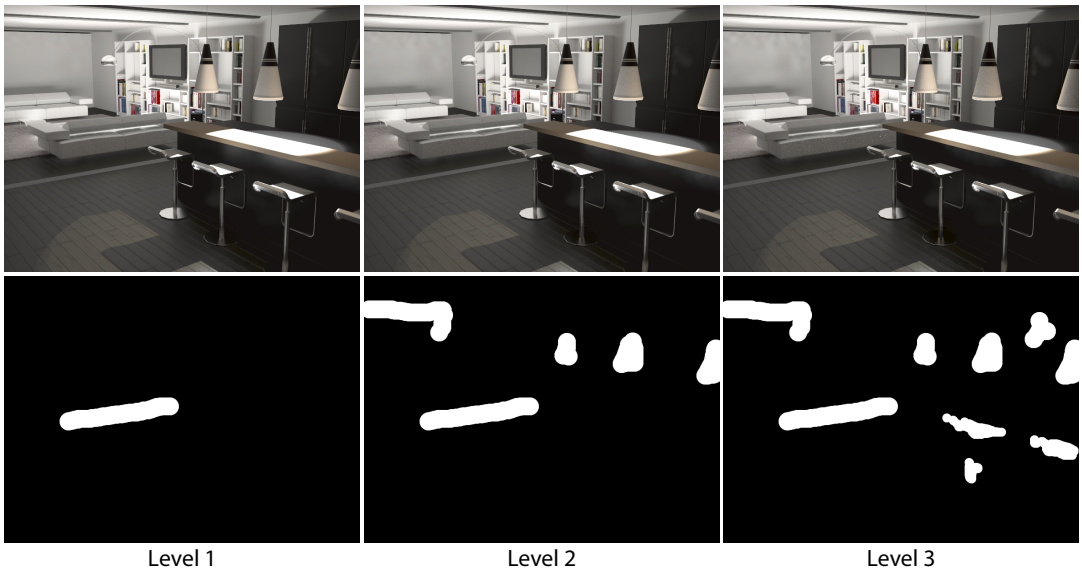


FIGURE 3.13: *An example scene with three levels of distortion magnitude (top row) and the corresponding distortion markings (bottom row). The distortion level increases from left to right, which results in adding newly marked regions.*

Observers Different groups of observers were asked to complete each subset of the dataset. At least 15 and at most 20 observers completed each subset. In total, 46 observers (aged 23 to 29) were recruited from among computer science students and researchers. The observers were paid for their participation. All observers had normal or corrected-to-normal vision and were also naive to the purpose of the experiment. To reduce the effect of fatigue, the experiment was split into several sessions, each lasting less than one hour. The post-experiment interviews indicated that the session length was acceptable and did not cause excessive fatigue.

3.4 Modeling Experimental Data

When fitting a visibility prediction model to the collected data, it is important to note that the data is the result of a stochastic process that is affected by noise and cannot be considered as the ground truth. The visibility threshold is not constant, as the detection can vary substantially between observers or even for the same observer when the measurements are repeated. Moreover, we collect binary data (visible or not) in a marking experiment, which is affected by more factors than the detection performance. If we had tried to fit the metrics directly to the data, we would have trained them to predict the data with all noise and irrelevant effects collected in the experiment. Therefore, we found it necessary to model the stochastic process and introduce such a model into the loss function used for calibrating the metrics.

But let us first consider what kind of process we want to model. Most of the existing visibility metrics, including VDP and HDR-VDP, attempt to predict a detection threshold for an average observer. This is done by measuring detection thresholds for each observer and then predicting the average of that data. The problem with this approach is that the visual performance of most observers is different from that of an average observer. For many applications, it is arguably more important to predict the performance of a population of observers rather than an average observer. Therefore, we aim to predict the proportion of the entire population that will perceive a difference. For that reason, we involve more observers (15–20) than typically found

in discrimination experiments (2–5), but we expect less precise measurements for each observer.

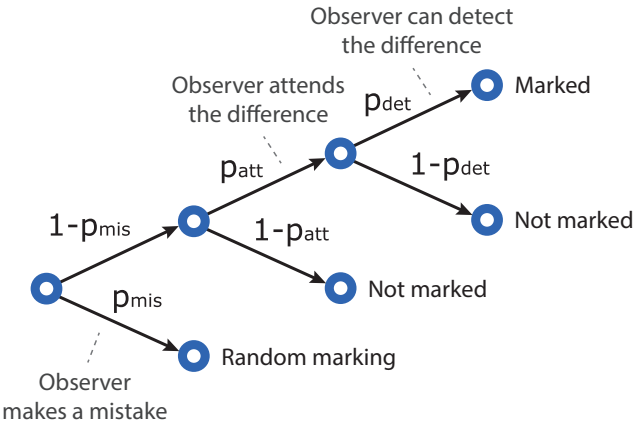


FIGURE 3.14: *The statistical process modeling observed data, given the probability of observer making a mistake (p_{mis}), the probability of attending (p_{att}) and detecting (p_{det}) differences in images.*

Let us model the likelihood of observing collected marking data given that we know the true probability of detection (p_{det}). We will later use such likelihood as a loss function. Firstly, we need to account for observers making a mistake and marking or not marking pixels regardless of whether they see a difference. We denote the probability of a mistake as p_{mis} and assume that markings are due to a mistake in 1% of the cases ($p_{mis} = 0.01$), which are mostly caused by the inaccurate marking of a selected region. Such a probability of a mistake is a common element of many statistical models of psychophysical procedures, which prevent strongly penalizing the outcome of an experiment if an observer did not act as expected. Secondly, we observed that many well-visible distortions were not marked because the observers were not attending a particular part of an image. Finding localized differences, particularly for computer graphics distortions, is a challenging task and not every detectable difference will be attended and spotted every time. Therefore, for a pixel to be marked, an observer needs to both attend a particular image location (the observer must look at the spot) and must be able to detect the difference (the difference must be visible). Given that the probability of attending an image location is p_{att} and the probability of detecting a difference is p_{det}, the entire process of marking for a single observer can be modeled statistically, as shown in Figure 3.14. If we have multiple observers marking an image location, the probability of observing an outcome of an experiment in which k out of N observers mark a patch is described by the Bernoulli process with an adjustment for the mistakes:

$$\begin{aligned} P(data) &= p_{mis} + (1 - p_{mis}) \binom{N}{k} (p_{att} \cdot p_{det})^k (1 - p_{att} \cdot p_{det})^{n-k} \\ &= p_{mis} + (1 - p_{mis})\, Binomial(k, N, p_{att} \cdot p_{det}) \,. \end{aligned} \tag{3.1}$$

In most practical applications we are mostly interested in predicting p_{det} – the probability that the difference is visible to an observer assuming that he is paying attention to every part of an image. However, we cannot infer p_{det} without knowing p_{att}. Even worse, p_{att} is likely to differ between observers, distortion types, and images. Therefore, it is a random variable rather than a constant.

However, we can estimate the distribution of p_{att}. We found that the distribution of p_{att} depends mostly on the type of the artifact. Therefore, we split our data into

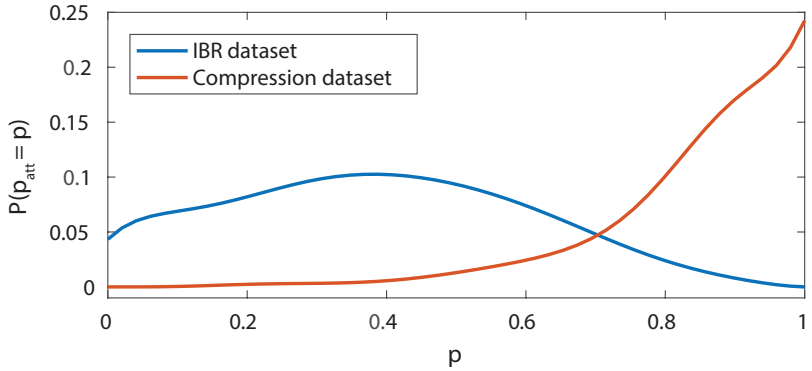


FIGURE 3.15: *The probability that the probability of attending a difference is equal to p, plotted separately for two datasets.*

datasets with similar distortion types and estimate p_{att} for each one. In every dataset, there are some image parts with very large differences in pixel values between distorted and reference images. If the difference is large enough (20/255 for our datasets), we can assume that the difference will always be detectable when observed. Since this corresponds to $p_{det} = 1$, p_{att} is distributed as:

$$P(p_{att} = p) = p_{att}(p) = \frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} Binomial(k(x,y), N, p) \,, \tag{3.2}$$

where Ω is a set of all pixels (x,y) with large pixel value differences and $|\Omega|$ is the cardinality of that set. For simplicity, we ignore p_{mis} in the above estimate. An example distribution of p_{att} for two datasets is plotted in Figure 3.15. Now we can incorporate the random variable $p_{att}(p)$ into our statistical model of the marking process and compute the log-likelihood that a set of p_{det} values predicted by a metric describes the marking data collected in the experiment:

$$\begin{aligned} L = \sum_{(x,y)\in\Theta} & \log[p_{mis} + (1 - p_{mis}) \\ & \cdot \int_0^1 p_{att}(p) \cdot Binomial\left(k(x,y), N, p_{att}(p) \cdot p_{det}(x,y)\right)\, dp] \,, \end{aligned} \tag{3.3}$$

where Θ is the set of all pixels with coordinates x, y. The second line of the equation is the expected value of observing the outcome given the distribution of p_{att}. Equation 3.3 gives a probabilistic loss function, which we use when fitting the visibility models.

To better understand the importance of modeling the probability of attending (p_{att}), let us consider the expected likelihood value (the integral in Equation 3.3) for two different datasets and for a single pixel instead of the entire image. The plots in Figure 3.16 show the probability that p_{det} is equal to a certain value if exactly k observers out of 20 have marked the pixel. It can be noted that when $k = 10$ observers mark the pixel in the IBR dataset (upper plot), the probability of detection can range from about 0.5 to 1. This means that if only half of the observers mark a pixel, the difference could still be perfectly detectable ($p_{det} = 1$) with high likelihood and that the lower number of markings could be attributed to the difficulty in spotting the differences ($p_{att}\approx 0.5$). In the compression dataset (lower plot), the probability for $k = 10$ is concentrated around $p_{det} = 0.55$, suggesting that almost half of the observers could not detect the difference even if they attended the corresponding spot in the

image.

Instead of fitting quality metrics directly to the noisy raw data, the likelihood function from Equation 3.3 lets us fit the probabilistic estimate of what the true detection thresholds are most likely to be. Furthermore, we do not assume a single estimate of the true detection probability (e.g. mean, mode, or expected value), but the distribution of such probabilities, accounting for uncertainty in the data, for example, due to a limited number of observers.

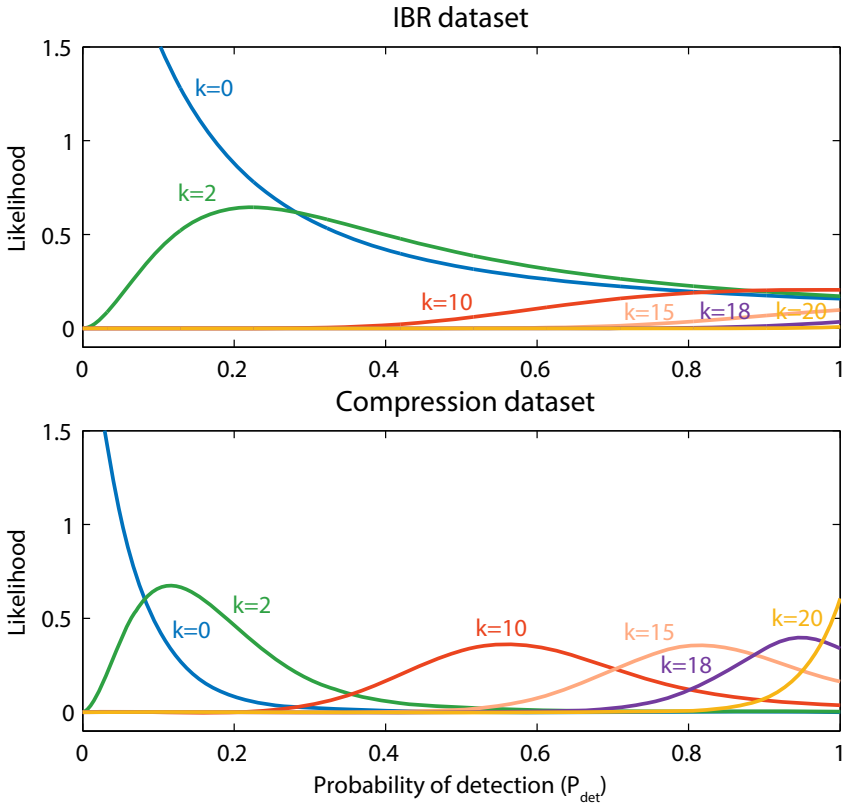


FIGURE 3.16: *The probability of detecting the difference for two datasets.*

3.5 Visibility Metrics

We adapt several popular image quality metrics to predict visibility and then use our dataset to train them. To train the metrics with a large number of images, we used a customized version of OpenTuner[1] optimization software. We extended the software so that it could be run on a compute cluster and computing metric predictions was distributed to a large number of nodes. Such parallelization was necessary for more complex metrics, such as HDR-VDP or SSIM. We calibrated the following metrics (refer to Section 2.2.1 for a short description of each metric):

[ABS] The absolute differences (D) between pixel values were divided by the threshold value t and then put into an expression for a psychometric function:

$$p_{det}(x,y) = 1 - \exp\left(\log(0.5)\cdot\left(\frac{D(x,y)}{t}\right)^{\beta}\right), \tag{3.4}$$

[1] `http://opentuner.org`

where x, y are pixel coordinates. The two optimized parameters were t and β. The absolute difference D was computed between luma values of distorted and reference images. The predicted $p_{det}(x,y)$ values can be used with Equation 3.3 to compute the probabilistic loss function.

[SSIM] We found that the values predicted by the SSIM metric are non-linearly related to the magnitude of visibility and benefit from the transformation:

$$D_{SSIM}(x,y) = \frac{1}{\epsilon}\left(\log\left(1 - M_{SSIM}(x,y) + \exp(-\epsilon)\right) + \epsilon\right) , \tag{3.5}$$

where $\epsilon = 10$ and M_{SSIM} is the original SSIM difference map. The transformation makes the D_{SSIM} values positive, in the range 0–1 and increasing with higher image differences. The $D_{SSIM}(x,y)$ values are then processed by the psychometric function from Equation 3.4. The optimized parameters were t, β and two parameters of the SSIM metric, C_1 and C_2 (refer to Equation 3.13 in original SSIM publication [Wang and Bovik, 2006]).

[VSI, FSIM] After transforming the difference maps D_{VSI} and D_{FSIM} into increasing values in the range 0–1, Equation 3.4 can directly be applied. The fitted parameters were t, β, and three parameters of the VSI metric, C_1, C_2, and C_3 (refer to Equations 4–6 in [Zhang et al., 2014]), or respectively two parameters of the FSIM metric, T_1 and T_2 (refer to Equations 4–5 in paper by Zhang et al. [2011]).

[CIEDE2000] The distorted and reference images were transformed into linear XYZ space assuming Rec. 709 color primaries and using a gain-gamma-offset display model simulating our experimental display. The predicted ΔE were transformed into probabilities using the psychometric function (Equation 3.4). The calibrated parameters were t and β.

[sCIELab] Our adaptation of sCIELab was identical to the one we used for CIEDE2000, except that the metric was also supplied with the image angular resolution in pixels per visual degree.

[Butteraugli] In the original Butteraugli implementation, the threshold for visible distortions is determined by a constant "good_quality". However, we found that this constant does not correlate well with human experiment results, and better results can be achieved if the map is transformed by the psychophysical function from Equation 3.4.

[HDR-VDP] We modified HDR-VDP (v2.2) to better predict our datasets. Firstly, we observed that including orientation-selective bands did not improve predictions for any of our datasets; therefore, we simplified the multi-scale decomposition to all-orientations spatial-frequency bands. Secondly, we improved spatial probability pooling. The original HDR-VDP was calibrated to detection datasets in which one distortion was visible at a time. This enabled using a simplified spatial pooling, in which all differences in an image were added together. However, it resulted in inaccurate results for our datasets, in which distortions vary in their magnitude across an image. We replaced the original pooling with spatial probability summation

$$P_{sp}(x,y) = \mathbf{1} - \exp\left(\log(1 - P(x,y) + \epsilon) * g_\sigma\right)(x,y) , \tag{3.6}$$

where $P(x, y)$ is the original probability of detection map (Equation 20 in paper by Mantiuk et al. [2011b]), ϵ is a small constant, and g_σ is the Gaussian kernel. The fitted parameters were the peak sensitivity, a self masking factor (`mask_self`), a cross-band masking factor (`mask_xn`), the p-exponent of the band difference (`mask_p`), and the standard deviation of the spatial pooling kernel (`si_sigma`) in visual degrees.

In the following sections, we add the prefix "T-", e. g., T-Butteraugli, to distinguish between the metrics trained by us and their original counterparts.

3.6 CNN-Based Metric

Inspired by the success of convolutional neural networks (CNNs) in many applications, we designed a fully convolutional architecture as a visibility metric and trained it on our dataset.

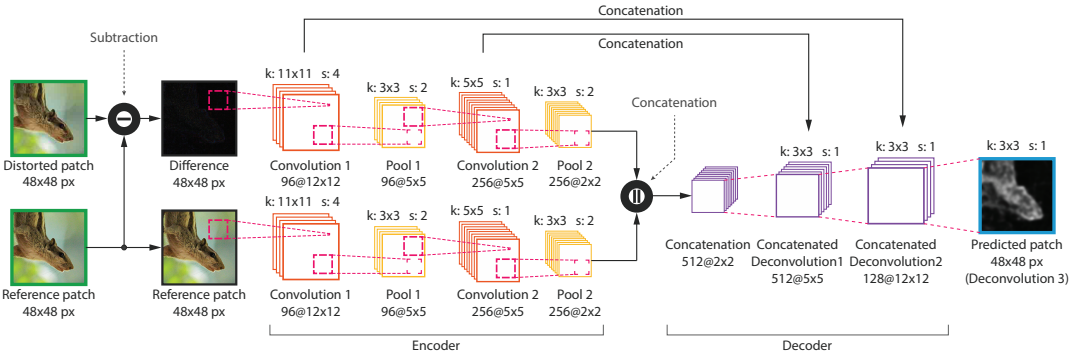


FIGURE 3.17: *Two-branch fully convolutional CNN architecture with the difference branch. The difference branch takes a difference between the distorted and reference images as the input, while the other branch accepts the reference image. The output is a visibility map, achieved by regression, with the same size as the input patch. Each branch contains two convolution layers with* 11×11 *kernel and stride 4 followed by another layer with* 5×5 *kernel and stride 1. The deconvolution section uses convolution layers with* 3×3 *kernel and stride 1.*

3.6.1 Two-Branch Fully Convoluted Architecture

Siamese networks gained popularity among tasks that involve difference comparisons or a relationship between two images. A Siamese CNN consists of two identical branches that share weights but have distinct inputs and outputs. For example, Bosse et al. [2016a] use Siamese architecture to encode distorted and reference patches in the features space, take the difference in that space and input it to the fully-connected layers to predict per-patch quality. After having experimented with such an architecture, we found that better performance can be achieved when a) the difference between distorted and reference patches is computed in image space rather than feature space, and b) fully-connected layers are replaced with convolutional layers.

Figure 3.17 illustrates our modified architecture, in which one branch is responsible for encoding the difference between distorted and reference patches and the other for encoding a reference patch. Unlike the Siamese architecture, our approach uses separate weights for each branch, as each branch encodes different information. Both branches are concatenated to preserve all features. The visibility map is reconstructed

from the feature-space representation using 3 deconvolution layers. Using convolutional layers instead of fully connected layers reduced the number of parameters and improved the ability of the metric to generalize to different datasets.

Formally we denote R as the reference patch and D as the distorted patch. We also define two mapping functions $F_{w^d_{conv}}$ and $F_{w^r_{conv}}$, where w^d_{conv} and w^r_{conv} represent the weights for convolutional layers of the difference branch and the respective weights for the reference branch. In addition, we use a mapping function $F_{w_{dec}}$ where w_{dec} represents the weights for deconvolution layers with the skip connection mechanism. Our metric $M_w(D, R)$ is formulated as:

$$M_w(D, R) = F_{w_{dec}}(Concatenate(F_{w^d_{conv}}(D - R), F_{w^r_{conv}}(R))) \tag{3.7}$$

Convolutional Layers Since our training dataset does not contain the necessary amount of images to perform a full training, we perform a fine-tuning of our network by initializing the weights via transfer learning from AlexNet implementation [Krizhevsky et al., 2012]. We found that the feature maps generated with two convolutional layers achieve similar results as with 5 layers, and consequently, we remove the last 3 convolutional layers. The two convolution layers alternate with pooling layers. We use the rectified linear unit (ReLU) as the activation function. In addition, to avoid overfitting, we set the dropout value to 0.5.

Deconvolutional Layers The concatenation is followed by the deconvolution layers, which reconstruct the final visibility map. To prevent checker-board patterns, each deconvolution is performed as a sequence of upsampling and convolution operations. Such patterns are a common problem if deconvolution is realized as a transposed convolution. Further refinement is achieved by skip-connections, which concatenate feature maps from the convolution layers of the difference branch with the deconvolution layers. Such skip-connections create new paths inside the neural network and help to avoid the issue of vanishing or exploding gradients.

3.6.2 Training and Testing

The network is trained by minimizing the likelihood function from Equation 3.3. The images are split into patches of 48×48 pixels, without overlapping parts. We found that patches of that size better preserve high-frequency details in the visibility maps. To increase the size of the dataset and prevent overfitting, we augment the data by horizontal and vertical flips with rotations of 90, 180, and 270 degrees. We also ignore all the patches for which there is no difference between their distorted and reference versions. The total number of patches is approximately 400,000. We train the network in 50,000 iterations, with a learning rate of $1e^{-5}$ and a learning decay rate of 0.9. To speed up the training process, we use a mini-batch technique with a batch size of 48. The network is trained using Adaptive Moment Estimation (Adam), which calculates adaptive learning rates for the parameters. The CNN architecture is implemented in TensorFlow 1.4 [2]. We perform training and testing exploiting TensorFlow GPU support on an NVIDIA GeForce GTX 980 Ti.

To predict a visibility map for a full-size image, we split it into 48×48 patches with 42-pixel overlap, infer a visibility map for each patch, and finally assemble the complete map by averaging each pixel shared by the overlapping patches. Prediction for an 800×600 pixel distorted image takes approximately 3.5 seconds.

[2] `https://www.tensorflow.org/`

3.7 Metric Results

To compare metric predictions, we computed the results for 5-fold cross-validation with an 80:20 split between training and testing sets. The split ensured that the testing set did not contain any of the scenes used for training, regardless of the distortion level. The results for all metrics, shown in Figure 3.18, indicate the CNN metric compares favorably to other metrics, where good performance can also be observed for T-Butteraugli and T-HDR-VDP. We did not find any evidence in our dataset that the color difference formula (T-CIEDE2000) offers an improvement over T-ABS computed between luma values. T-sCIELab performed slightly better than T-ABS, but for many datasets, they are quite comparable. It is worth noting that some datasets are more difficult to predict (lower likelihood) than others; notably, COMPRESSION was the most difficult dataset, followed by PERCEPTIONPATTERNS.

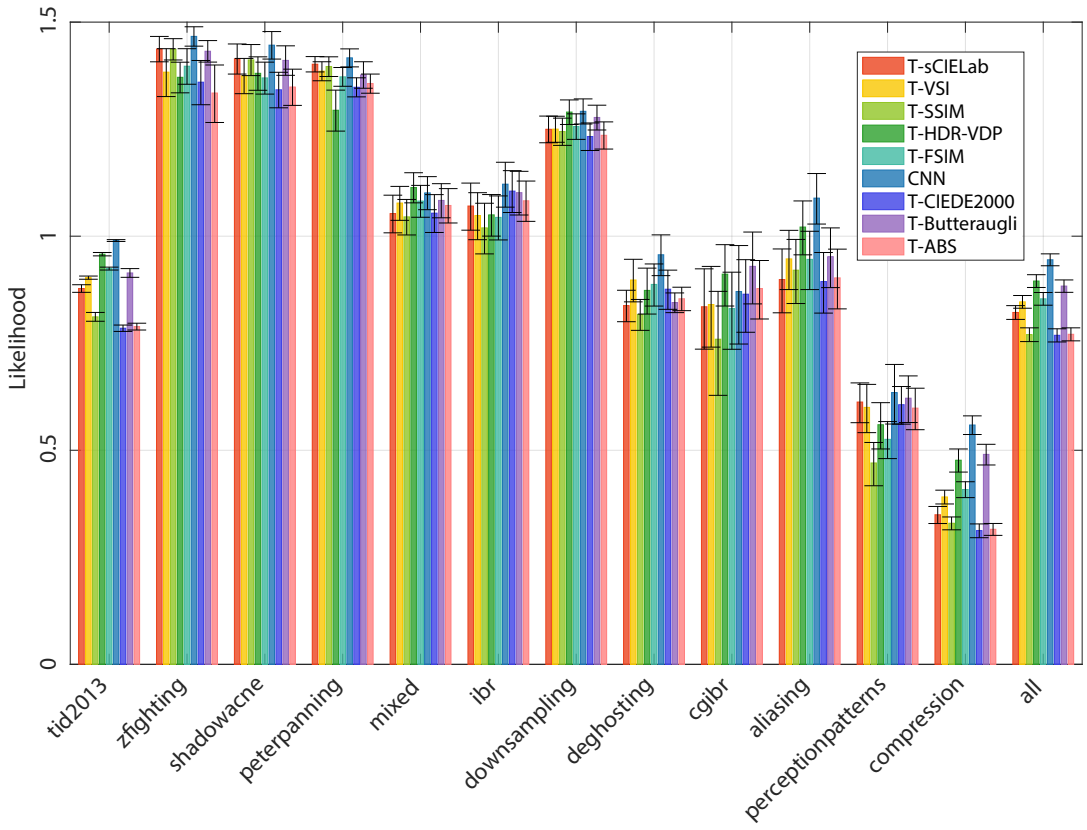


FIGURE 3.18: *The results of cross-validation of the quality metrics for each dataset and for all datasets together. The error bars denote standard error when averaging across images in the dataset.*

It must be noted that the metrics have been retrained and modified or extended to better predict our dataset. In Figure 3.19, we show the predictions of the original metrics compared to the predictions of the newly retrained metrics. In general, the original metrics were too sensitive and predicted mostly invisible distortions.

Figure 3.20 shows metric predictions for a few selected interesting cases from our dataset. The complete set of predictions, as well as the code and the parameters of the retrained metrics, can be found on the project webpage[3]. Image *uncorrelated noise* contains three noisy circular patterns modulated by a Gaussian envelope, presented on the background of a lower amplitude noise. The reference image contains only the background pattern, without the circular patterns, but the random seed used

[3] `https://visibility-metrics.mpi-inf.mpg.de`

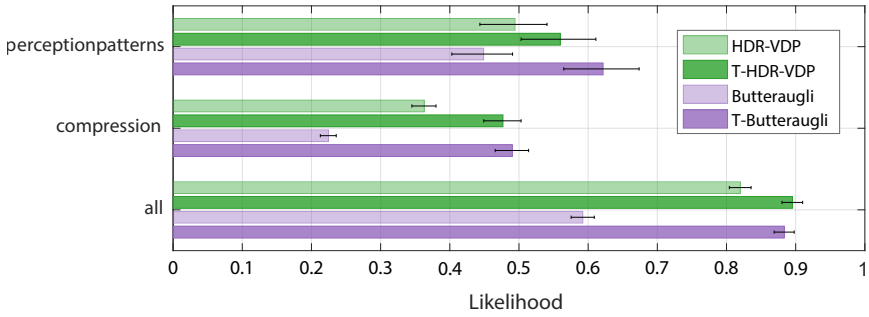


FIGURE 3.19: *Improvement in prediction performance between the original and retrained metrics. For brevity, only two selected subsets and overall performance are reported.*

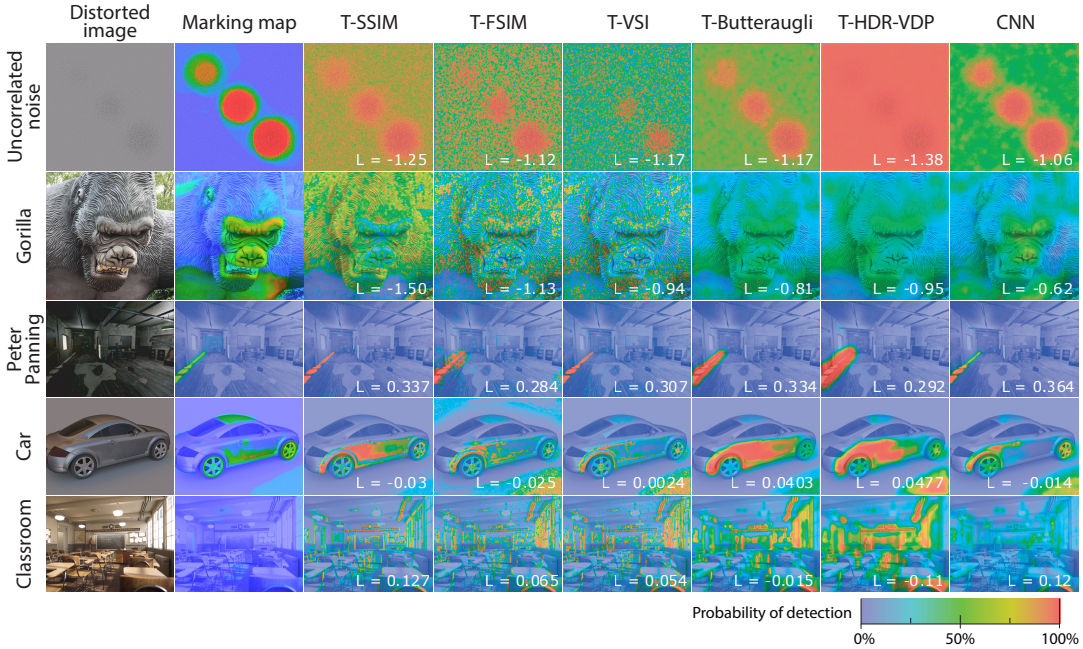


FIGURE 3.20: *Distorted images, observers' markings, and metric predictions for a few selected images from the dataset. Metric predictions must be viewed in color. Numbers present on each predicted visibility map indicate the likelihood value (a higher value indicates better accuracy).*

to generate the background noise was different than in the distorted image. While observers could not spot the difference in the background noise pattern in a side-by-side presentation, such a difference triggered detection for all metrics. T-Butteraugli and CNN performed better, partially ignoring the difference in background noise.

The *gorilla* image was distorted by image compression. It contains complex masking patterns, which modulate the visibility of the distortions. The simple metrics, such as T-SSIM, T-FSIM, and T-VSI, failed to predict such masking. More advanced metrics, CNN, T-Butteraugli, and T-HDR-VDP, were more accurate in indicating the higher visibility of distortions on the gorilla's face while the chest area was marked correctly only by CNN.

The *peter panning* image contains distortion caused by shadow mapping where the shadow is detached from an object casting it [Piórkowski et al., 2017]. The images also contain small differences in pixel values due to shading and post-processing effects in the game engine. T-FSIM metric failed to mask such small differences (best seen in the electronic version), while other metrics correctly ignore the visibility of those small differences. T-Butteraugli and T-HDR-VDP tend to excessively expand

the region with the difference.

The *car* image contains distortion on the body of the car, but there is also a readily visible noise pattern in the bottom right corner. Very few observers marked that noise pattern in the corner, as most people were paying attention to the car. This is an example of a case in which we cannot treat observers' markings as ground truth and we need the statistical model from Section 3.4 to correctly model uncertainty in the data. It is worth noting that all metrics correctly predicted the visibility of the noise pattern.

The *classroom* image contains a rendering of the same scene but from a slightly different camera position in the distorted and reference images. While the observers could not notice any differences, such pixel misalignment triggered a lot of false positives for most metrics. CNN could only partially compensate for pixel misalignment.

3.8 Applications

In this section, we present three applications that demonstrate the utility of the best-performing visibility metrics. In Section 3.8.1 we investigate visually lossless JPEG compression controlled by the metrics. In Section 3.8.2 the metrics are used to determine the maximum subsampling level for a single-image super-resolution. In Section 3.8.3 we use CNN metric to adjust content-dependent watermarks so that their intensity is maximized while they remain imperceptible.

3.8.1 Visually Lossless Image Compression

The objective of visually lossless image compression is to compress an image to the lowest bit rate while ensuring that compression artifacts remain invisible. JPEG quality setting is rarely manually fine-tuned and most images are saved using the default quality setting, which is typically set to 90 (out of 100) in a majority of software, producing much larger files than needed. In this section, we show that visibility metrics can be used to automatically find the JPEG quality setting for visually lossless compression, which can significantly reduce the file size.

To validate metrics' performance in this application, we conducted an additional experiment, in which observers indicated the lowest JPEG quality setting for which distortions remained invisible in a side-by-side presentation. In the experiment, we use the standard JPEG codec (libjpeg[4]). To avoid using the same images for training, we used Rawzor's free dataset[5] which contains a rich set of image content. The images are cropped to 960×600 pixels to fit on our screen. The images are distorted by compression with a standard JPEG codec using a range of compression qualities. The experimental procedure involved selecting a distorted image from the four presented, where only one image was distorted (four-alternative-forced-choice protocol). The quality setting was adaptively adjusted using the QUEST method. Between 20 and 30 trials were collected per image to find the quality settings at which an observer could select a distorted image with a 75% confidence probability level in the QUEST procedure. 10 observers completed the experiment. To predict the visually lossless quality setting for JPEG compression, we take the maximum value of the metrics' predicted visibility map, which gives a conservative estimate. A similar approach was in [Alakuijala et al., 2017]. We searched the quality settings from 0 to 98 and chose the lowest quality setting that produced the visibility map of a maximum

[4] `https://github.com/LuaDist/libjpeg`

[5] `http://imagecompression.info/test_images/`

value less than 0.5, which corresponds to 50% of observers spotting the difference. We select the best three metrics, CNN, T-HDR-VDP, and T-Butteraugli, and their original versions, HDR-VDP and Butteraugli, for evaluation.

The results of the experiment and the predictions of the top-performing visibility metrics are shown in Figure 3.21. The pink line denotes the median value computed across observers, and the pink shaded area represents the range between the 20^{th} and 80^{th} percentiles. CNN, T-HDR-VDP, and T-Butteraugli correlate reasonably well with the experiment results, although the distortions in scenes 9 and 11, were under-predicted by the trained metrics. The most visible distortions in those images are due to contouring in smoothly shaded regions. Such distortion types were missing in our training set, which could lead to worse-than-expected performance.

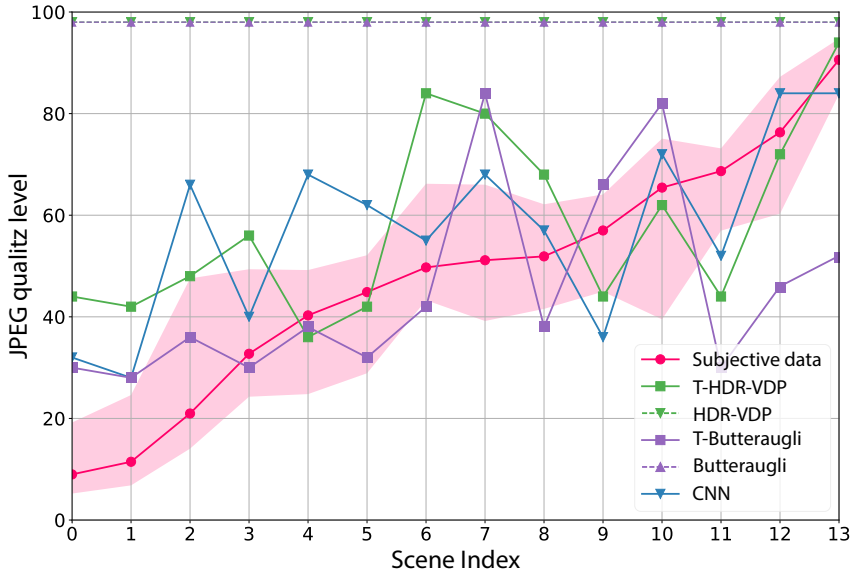


FIGURE 3.21: *Results for visual lossless image compression. The pink line is the median and the pink shaded region is 20^{th} and 80^{th} percentile of manual adjustment.*

We quantify the accuracy of metrics' predictions as the mean squared error (MSE) between the predictions and levels found in the experiment. Among the top three metrics, CNN's performance is the best with an MSE of 367.7 followed by T-HDR-VDP with an MSE of 467.5 and T-Butteraugli with an MSE of 479.4. The original (untrained) versions of HDR-VDP and Butteraugli resulted in strongly over-predicted visibility of JPEG artifacts. This result confirms that, with our proposed dataset, the trained metrics could generalize well to different distortion types (we did not include JPEG distortions in the dataset) and different content. Compared with the common practice of setting the quality to a fixed value of 90, the best CNN metric could help to reduce file size on average by 60% for the selected set of images.

3.8.2 Super-Resolution from Downsampled Images

Super-resolution (SR) methods reconstruct a higher-resolution image from a low-resolution (LR) image or images. In this section, we consider a single-image super-resolution, where the method is used to reduce image resolution in such a way that a visually indistinguishable full-resolution image can be reconstructed from a low-resolution image.

We consider three scenes generated with Arnold renderer[6] (two of them presented

[6] `https://www.solidangle.com/arnold/`

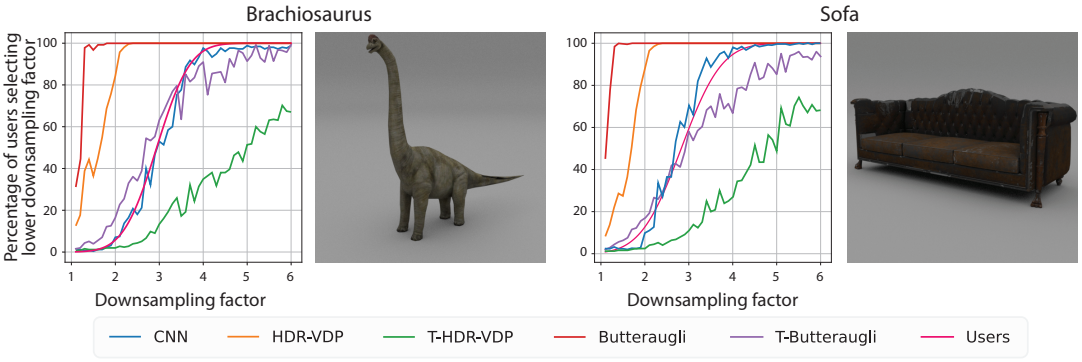


FIGURE 3.22: *Comparison of the metric performance in the super-resolution application. Two exemplar scenes are considered (Brachiosaurus and Sofa) for which the user data on the distortion visibility have been collected as depicted in pink in the corresponding graphs. The graphs show also the metric prediction of the user data for various downsampling factors.*

in Figure 3.22). The scenes depict objects on a quasi-uniform background with a small amount of noise caused by global illumination approximation. Simple scenes were selected to make it easier for observers to find the distortions in an experiment.

We employ the traditional projection onto convex set (POCS [Panda et al., 2011]) algorithm for reconstructing the SR image from a single LR image. We use a MATLAB implementation, generating LR images by applying downsampling factors from 1.1 to 6.0.

To validate the metric performance, we collected the user data on the distortion visibility for various downsampling levels with respect to the full-resolution reference. We ran the same 4AFC QUEST procedure as in Section 3.8.1. Between 20 and 40 QUEST trials were collected per scene and per observer to find the detection threshold for the level of the downsampling factor. 20 observers completed the experiment.

The pink line in Figure 3.22 shows the percentage of the population that selected lower downsampling than the one shown on the x-axis. The smooth shape of that line was estimated by fitting a normal distribution to the collected thresholds and then plotting the cumulative of that distribution. To plot metric predictions, we take the maximum value in the visibility difference map to account for the most visually critical distortion, similar to the JPEG application. To compare metric performance, we compute MSE between each metric prediction and the user data for all downsampling levels. After averaging the MSE values for all three scenes, CNN resulted in the smallest error of 19.95, followed by T-Butteraugli with an error of 96.96.

3.8.3 Content-Adaptive Watermarking

Watermarking is the technique of changing the signal in order to embed information about the data. For visual media, the watermark is usually a logo or text superimposed on an image that remains imperceptible. Our application is inspired by watermarking techniques and its objective is to show that our metric is able to detect correctly the areas where contrast masking lowers the visibility of a watermark. Contrast masking is an important characteristic of the HVS that results in the reduced visibility of a distortion that is imposed on the image pattern with similar spatial frequencies and orientations [Mantiuk et al., 2011b; Chandler, 2013]. Contrast masking becomes stronger with increasing contrast of the masker, which we employ to increase the watermark intensity. More information on masking can be found in Section 2.1.6.

To add a watermark, we add a grid of small gray-shaded watermark patches (64×64 pixels) to the reference image. We start with high-intensity watermarks so that the differences are clearly visible. Then, we employ the CNN metric to generate the distortion visibility map and to determine which watermark patches result in visible differences (we use a custom threshold of 6%). The intensity of all watermark patches in which at least one pixel contains visible difference is reduced by 1 (the minimum step). This process is repeated until the metric does not detect any visible difference. Note that this procedure is equivalent to pooling the maximum value per patch, rather than per image, as it was done for JPEG and super-resolution applications.

Although our metric was not trained for this particular distortion type, the watermarks optimized by our metric remain imperceptible in a side-by-side comparison with their references (Figure 3.23). This result demonstrates that our metric can deal with distortion types not present in the training dataset. For a better illustration, we also provide the watermarks multiplied by a constant, which show that our metric is able to correctly detect the areas that mask distortions. For example, in the rocks area (top row) and city lights (bottom row), which contain high frequencies of relatively high contrast, visual masking becomes stronger and our metric allows for higher intensity of the watermark. The watermark intensity remains small in the smooth gradient regions.



FIGURE 3.23: *Examples of content-adaptive watermark application. Our metric applies a brighter watermark in high-frequency and contrast areas, like the rocks in the first example and city lights in the second example.*

3.9 Extension for Viewing Distance and Display Brightness

Although the CNN-based visibility metric we proposed in Section 3.6 exhibits exceptional performance and applicability, it is constrained to a specific viewing condition based on the dataset it was trained on. This fixed viewing condition corresponds to a display with a peak luminance of 110 cd/m^2 and an angular resolution of 40 pixels per visual degree (ppd).

In this section, we propose an extension to our CNN-based metric that enables it to accommodate a range of display brightness levels and angular resolutions. To achieve this, we first obtain the necessary training data by utilizing an improved version of HDR-VDP (improvement method can be found in Section 3.5) for the generation of predictions for a large set of images affected by JPEG and WebP image

compression. In addition, we collect a small set of human-labeled data for fine-tuning the metric (Section 3.9.1).

To achieve the new metric modality, we combine white-box models of luminance masking and spatial resampling with our CNN-based architecture previously mentioned in Section 3.6. Description of this process and the training procedure can be found in Sections 3.9.2 and 3.9.3, respectively. Finally, in Section 3.9.4, we validate the performance of a new approach.

The newly-collected *LocVisVC* dataset[7] of 264 images labeled under different viewing conditions and metric code[8] are publicly accessible online.

3.9.1 Data Collection

In this section, we describe all necessary modifications in the data collection process to account for various viewing conditions, including different display peak luminance levels and viewing distances.

Stimuli and Experimental Procedure We conducted a random selection of 66 images from the LocVis dataset (Section 3.2). These chosen scenes encompassed various types of distortions, including compression, synthetic perception patterns, and artifacts arising from image-based rendering methods. Each selected scene exhibited up to three levels of distortion, such as different amplitudes of noise or varying compression levels. To ensure consistency, we followed the same experimental procedure outlined in the previous chapter. For further information, please refer to Section 3.3.

Observers and Experimental Setup The subjective study involved the same group of observers mentioned in Section 3.3. We also utilized the same experimental setup while making adjustments for varying viewing conditions. To address the issue of various luminance levels, we employed a Minolta LS100 luminance meter to calibrate the screen for two distinct peak luminance levels: $10\,\mathrm{cd/m^2}$ and $220\,\mathrm{cd/m^2}$. In order to achieve a luminance of $10\,\mathrm{cd/m^2}$, the display was dimmed, and a 0.6 Neutral Density (ND) filter, which reduces the light intensity by a factor of 4, was applied to the screen. These two configurations encompass the range of luminance typically found in most displays[9]. Furthermore, we varied the viewing distance, setting it to 40 cm and 86 cm, which correspond to angular resolutions of 30 and 60 pixels per visual degree, respectively.

Results Figure 3.24 illustrates the trends of visibility changes under different luminance and distance conditions. With the increase of luminance and the decrease of angular resolution, distortions become more visible, which agrees with empirical observations and previous research [Mantiuk et al., 2011a].

Pre-Training Dataset As the manually labeled datasets were insufficient for training, we also prepared a dataset with synthetic labels generated with the HDR-VDP visibility metric. We used 200 high-quality photographs obtained directly from camera RAW files. The images were distorted by encoding and decoding using JPEG[10] and WebP[11] image compression at the quality settings of 20, 50, and 90. We then

[7] https://doi.org/10.17863/CAM.37996
[8] https://github.com/ynyCL/DPVM
[9] https://www.laptopmag.com/benchmarks/display-brightness
[10] https://github.com/LuaDist/libjpeg
[11] https://developers.google.com/speed/webp

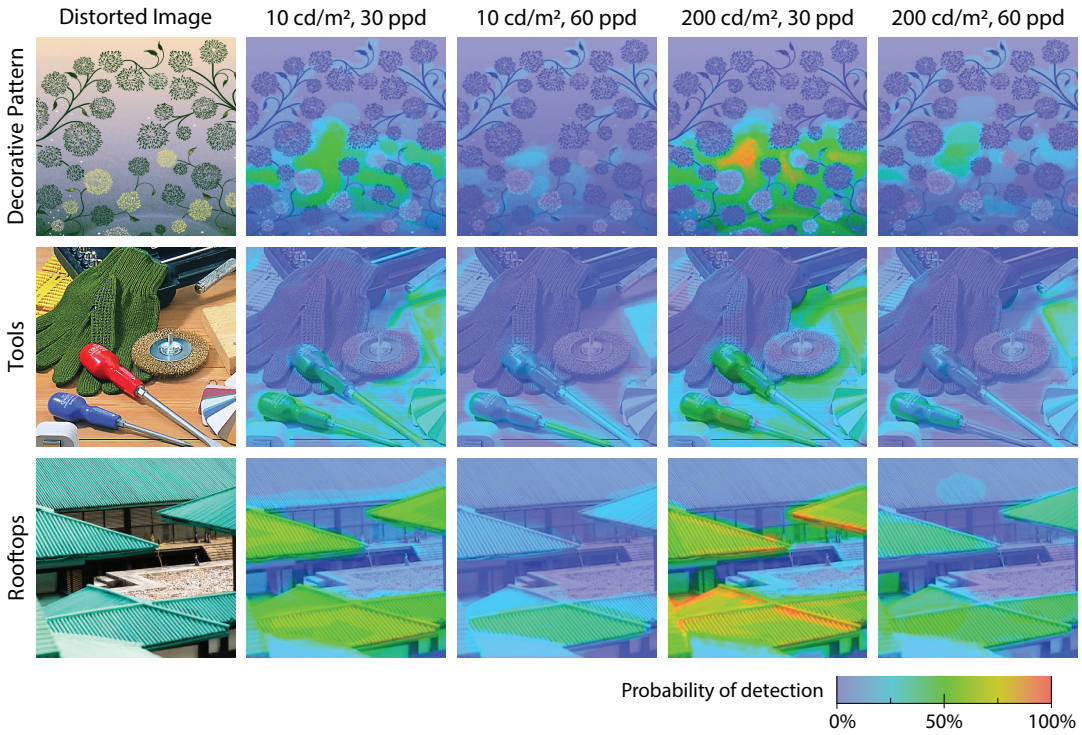


FIGURE 3.24: *Examples of images and subjective data from the LocVisVC dataset. Decreasing ppd (decreasing the distance between the observer and the display) for the same luminance condition increases the visibility of artifacts. When luminance is increased while keeping the same ppd condition, the visibility of artifacts also increases.*

randomly selected 50 images as the base scenes for our dataset. Each of these images was converted into linear colorimetric units using the display model (explained in Section 3.9.2), assuming the peak luminance of 10 cd/m^2, 110 cd/m^2, and 220 cd/m^2. The visibility map for these images was then predicted for the angular resolutions of 30, 40, 50, and 60 pixels per visual degree, resulting in a total of 600 labeled images. A summary of the dataset can be found in Table 3.2.

TABLE 3.2: *The subsets of the dataset used for training.* LOCVIS *is the original dataset described in Section 3.2,* VIEWCOND *is the newly measured LocVisViewCond dataset, and* PRETRAIN *is the HDR-VDP generated synthetic dataset for pre-training.*

Subset name	Scen.	Imag.	Levels	Level generation method	Peak lum.	ppd
LOCVIS	170	557	1-3	various methods	100,110 cd/m^2	40, 60
VIEWCOND	26	264	1-3	various methods	10, 200 cd/m^2	30, 60
PRETRAIN	200	600	3	JPEG and WebP compression	10, 110, 200 cd/m^2	30:10:60

3.9.2 Metric Architecture

In order to account for various viewing conditions, certain adjustments are required in the architecture presented in Section 3.6. Overcoming the obstacle of limited visibility data, we propose a hybrid architecture that explicitly incorporates the display peak luminance and viewing distance as part of a preprocessing step. In this section, we introduce each essential element of the new design and outline all the modifications.

Display Model

Since modern displays differ substantially in their peak brightness, it is important to model how much light they emit. As an example, some mobile displays can reach the peak luminance of 900 cd/m^2 and can be dimmed to as low light levels as 3 cd/m^2. The visibility of image distortions is very different between both cases. To model the amount of emitted light, we use the standard gain-gamma-offset display model:

$$L = (L_{\text{peak}} - L_{\text{black}}) \left(\frac{I}{255} \right)^{2.2} + L_{\text{black}}, \tag{3.8}$$

where I is the input pixel value, L_{peak} is the peak luminance of the display, and L_{black} is the luminance of black level (light emitted from pixels set to black). Each image provided to the metric is first transformed from pixel values to colorimetric red, green, and blue values using the display model from the equation above.

Luminance Masking

Since differences are less visible at lower absolute luminance levels, we need to account for this drop in visual system sensitivity. Luminance masking can be modeled by a transfer function derived from the contrast sensitivity function of visual system [Mantiuk et al., 2011a; Aydin et al., 2008a]. The transfer function we use is also known as Perceptually Uniform (PU) encoding [Aydin et al., 2008a], as it transforms physical luminance into approximately perceptually uniform units. The PU encoding is defined as an integral of the inverse of detection thresholds:

$$P(L) = \int_{L_{min}}^{L} \frac{1}{T(l)} dl \tag{3.9}$$

where L_{min} is the minimum luminance to be encoded. The detection thresholds $T(L)$ are modeled as a function of absolute luminance L:

$$T(L) = S \cdot \left(\left(\frac{C_1}{L} \right)^{C_2} + 1 \right)^{C_3} \tag{3.10}$$

Where S is the absolute sensitivity constant, L is the luminance, and C_1, C_2, and C_3 are parameters obtained by fitting to contrast sensitivity measurements. We use the parameters from [Mantiuk et al., 2011a].

For comparison, we also experiment with the logarithmic encoding of luminance, as it is the first-order approximation of the visual system response, which accounts for the Fechner law. We show both perceptual encoding functions in Figure 3.25.

Viewing Distance

An intuitive way to account for the viewing distance is to provide the model with an image that maintains a constant angular resolution. Doing so ensures that the spatial frequencies remain unchanged regardless of the distance at which the image is viewed. The angular resolution of an image can be computed as:

$$r = \frac{N_y}{h_{deg}} \quad [ppd], \tag{3.11}$$

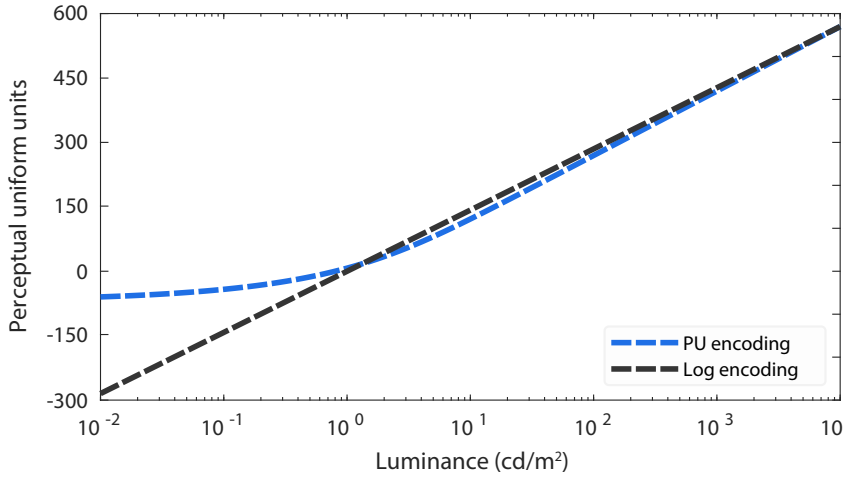


FIGURE 3.25: *PU and logarithmic transform functions for converting absolute light levels into approximately perceptually uniform values, which could be input to a CNN.*

where N_y is the display's vertical resolution expressed in pixels. h_{deg} is the display height expressed in visual degrees and can be computed using the following equation:

$$h_{deg} = 2 \arctan\left(\frac{h_{mm}}{2\,d_{mm}}\right), \tag{3.12}$$

where d_{mm} is the viewing distance expressed in millimeters, and h_{mm} is the display height expressed in millimeters. Once we know the angular resolution of the input image, we resample it so that it has the angular resolution of 60 ppd. This is the highest resolution in our dataset and also a reasonable limit for most visual tasks since the sensitivity of the visual system drops rapidly below 30 cycles-per-degree (cpd) [Barten, 1999].

Since resampling alone cannot account for all frequency-dependent effects, such as the shift of peak sensitivity with luminance, we also concatenate the *ppd* parameter to the latent code. This is achieved by concatenating a slice with replicated *ppd* values to the feature maps generated by the encoders (see Figure 3.26).

Architecture Modification

We introduce the above-mentioned modifications into the existing architecture, as shown in Figure 3.26. First, the display model and perceptual transform are applied to both the distorted and reference images. Next, these images are cropped and resampled to patches of resolution 48×48 pixels. Such input is fed to the neural network described previously in Section 3.6. However, additional input in the form of a slice representing angular resolution, expressed in pixels-per-degree (ppd), is added at the concatenation step.

3.9.3 Training

For training, we use the proposed in Section 3.6 setup of the loss function, optimizer, and hyperparameters. However, this time we split the training process into two stages.

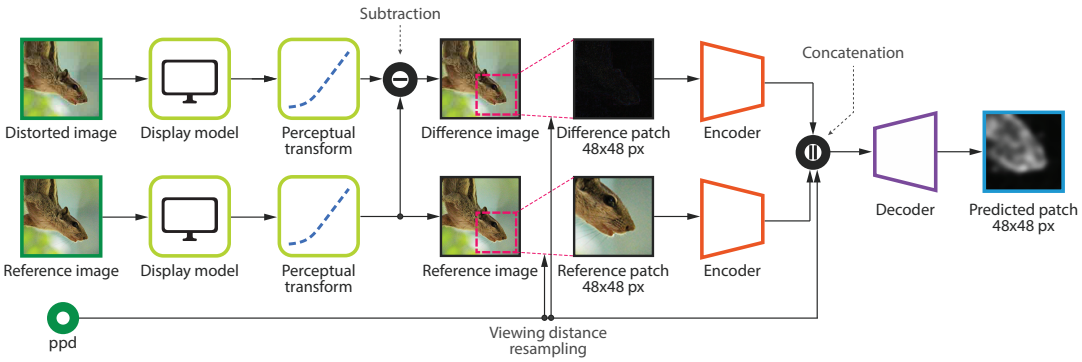


FIGURE 3.26: *Architecture of the viewing-condition-aware CNN visibility metric. The metric described in Section 3.6 has been modified by adding a display model and perceptual transform to account for the display brightness. Moreover, the viewing distance resampling and concatenation with the angular resolution, expressed in pixel-per-degree (ppd), at the bottleneck have been added to introduce varying viewing distances. Please note that the encoder and decoder parts of the network remained unchanged and were collapsed for clarity of presentation.*

Stage 1: Pre-Training with HDR-VDP As the collected dataset contains only limited variation in viewing distance and display peak luminance levels, we supplement our training with over 13 million patches that have been automatically labeled by a white-box visibility metric — HDR-VDP. The generation of this PRETRAIN dataset was explained in Section 3.9.1. The idea is inspired by the work of Kim et al. [2017], who demonstrated that PSNR scores can be used to pre-train CNN-based quality metrics. Similarly, we run 20000 iterations of training on the PRETRAIN dataset, which is followed by fine-tuning in Stage 2. Although the labels generated by HDR-VDP can be inaccurate, they capture the general relationship between input and output patches and therefore prime the CNN to capture the relationships, which could be missing in manually labeled data.

Stage 2: Fine-Tuning At this stage, we initialize the neural network with weights from the first stage and use the manually labeled datasets for training.

3.9.4 Results

To validate metric performance, we randomly split the LocVisVC dataset into five folds, ensuring each scene is in a single fold, and run a 5-fold cross-validation. We report the mean and standard error of the likelihood used for the loss function (the higher likelihood indicates the higher accuracy).

PU vs. Logarithmic Encoding First, we compare performance when either PU encoding or a logarithmic function is used to account for luminance masking. The likelihood for the PU encoding (0.877 ± 0.015) was substantially higher than for the logarithmic function (0.705 ± 0.02). This suggests that luminance masking is a significant effect in our dataset.

HDR-VDP Pre-Training Next, we examine the impact of pre-training on metric performance. Our approach involves conducting pre-training for a range of iterations, spanning from 10,000 to 50,000, which is followed by fine-tuning for 50,000 steps. The results, shown in Figure 3.27, demonstrate that pre-training consistently

improves accuracy. However, performance declines after approximately 20,000 iterations, emphasizing the importance of controlling the extent of pre-training to ensure effective learning from human-labeled data. For the final evaluation, we utilize 20,000 iterations of pre-training.

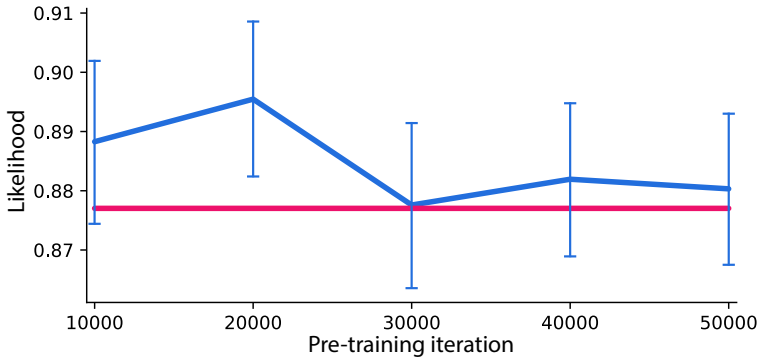


FIGURE 3.27: *The effect of pre-training iterations on the performance. The pink line denotes the result without pre-training. The error bars denote standard errors. A higher likelihood corresponds to better accuracy*

Metrics Comparison Finally, we compare the proposed DPVM metric to the HDR-VDP, which is the only visibility metric that can account for the viewing conditions. For a fair comparison, we retrain HDR-VDP-2.2 (for details on this process, please refer to Section 3.5) on the same dataset as used for the training of the CNN-based metric. The result of cross-validation is shown individually for each subset in Figure 3.28. The likelihood of the proposed DPVM metric is significantly higher for each subset, demonstrating the CNN-based metric can be trained with higher accuracy.

Examples of metric predictions and user markings are shown in Figure 3.29. We can observe there that similar to HDR-VDP, DPVM can account for the change of viewing distance and absolute luminance as shown in rows 1–3. Pre-training with HDR-VDP also helps improve the generalization performance for most cases.

3.10 Limitations

Our approach to data collection and modeling is not free from limitations. The experimental task, although intuitive and very efficient, leads to ambiguity between the ability to find and detect distortion. We address this problem by inferring the likely probability of observing a particular distortion. However, we also found that using smaller images, such as the 512×512-pixel crops used in the COMPRESSION set, greatly reduces the effect of visual search and allows us to collect more consistent data. Moreover, while the extension proposed in Section 3.9 makes the metric more flexible and allows to predict visibility in various viewing conditions present in the LocVisVC dataset, its performance in conditions with extremely low or high peak luminance levels of the display remains unclear.

3.11 Conclusions

Prediction of visible distortions in images is a challenging task, and no existing visibility metric can provide robust predictions. The challenge comes from the complexity of human visual perception but also from the lack of sufficiently large datasets which

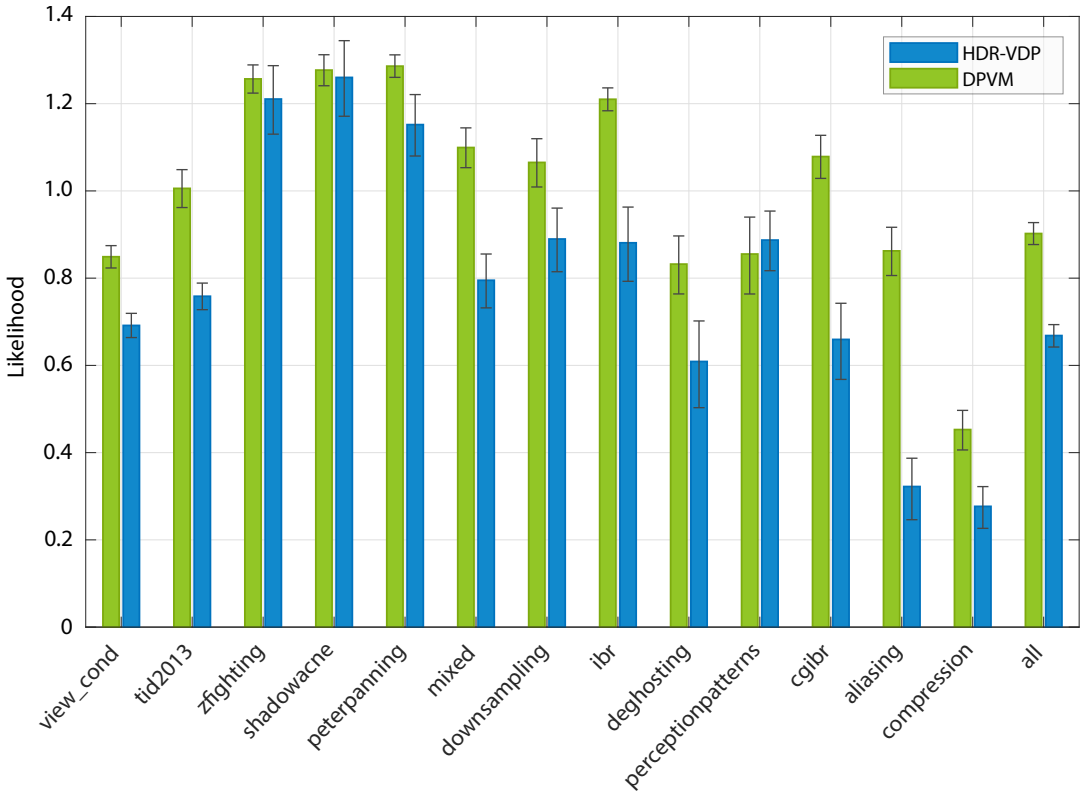


FIGURE 3.28: *Metric cross-validation results for each subset and for the entire dataset.*

could be used to train and validate visibility metrics. This work contributes towards creating such a large dataset. We propose to use an efficient experimental method, which, however, can misrepresent the true detection performance when the observers are not able to locate all visible distortions. For that reason, we create a statistical model that can describe the data's uncertainty and serve as a loss function for training metrics.

We use the dataset to train the existing visibility metrics and demonstrate that their performance can be significantly improved. Additionally, we develop a new CNN-based metric that surpasses the performance of current metrics, albeit with the constraint of being applicable only under specific viewing conditions. We remove this limitation by modeling explicitly the effects related to changes of viewing distance and display luminance. By implementing this modification and utilizing a newly collected dataset that covers a range of viewing conditions, we not only attain superior accuracy but also achieve a level of flexibility that is on par with the most advanced existing methods.

We also demonstrate the utility of our CNN metric in three practical applications: visually lossless JPEG compression, superresolution, and watermarking. In future work, we would like to investigate other applications, such as automatic simplification of 3D scene complexity (textures, geometry, shading) while ensuring no loss of quality or comparison of rendering methods while ignoring differences below the visibility threshold.

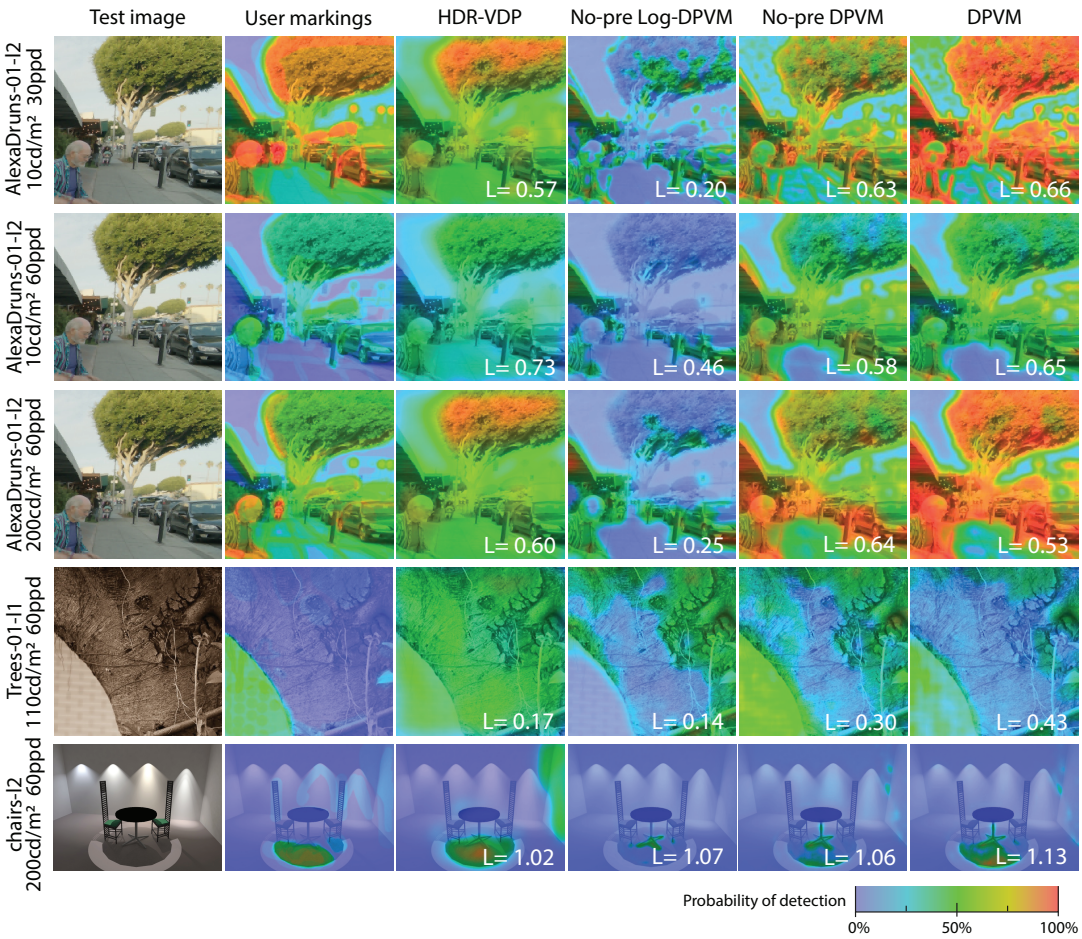


FIGURE 3.29: *Distorted images, users' markings, and metrics' predictions examples from the dataset. The L value represents the likelihood (higher values indicate better performance). The prefix* no-pre *in the metrics' labels indicates that the predictions were made without HDR-VDP pre-training.*

Chapter 4

Selecting Texture Resolution Using a Task-Specific Visibility Metric

4.1 Introduction

Since the beginning of computer graphics, the complexity of rendered scenes evolved from small environments with basic shading to realistic-looking large-scale virtual worlds that often require days or weeks to fully explore. One of the critical components of these scenes is textures storing local color information in addition to other attributes such as surface normals, geometry displacement or reflectance properties. Since all of them significantly contribute to the faithful reproduction of real-world appearance, their complexity as well as quality is crucial for the final experience provided to a user. To address these quality demands, graphics vendors develop more complex tools, such as physically based texture painter [Deguy et al., 2016] or terrain generation software [Carrier, 2018], which simplify the process of enriching textures with details, and therefore, increase the photo-realism of produced scenes.

Since one object may consist of many different parts and materials, the size of all textures can quickly increase with the complexity of the scene. As a consequence, many games released in recent years require a lot of memory space to store all the material attributes, e.g., Final Fantasy XV (170 GB) or Gears of War 4 (250 GB). Large textures also impact the overall performance by drastically increasing the loading time and memory bandwidth to GPU, and therefore, contributing to the reduced efficiency of a rendering engine. Textures size becomes an even more significant problem in the context of new ultra-resolution displays or virtual reality (VR) devices, where the quality demands with respect to spatial and temporal resolutions are pushed to the limits, or in the context of mobile devices, where the computational and memory budgets are very tight [Epic Games, 2018].

Texture compression and resampling are critical preprocessing steps which address the above problem. They can balance the quality and memory requirements by trying to keep the scene description in the GPU memory as much as possible while minimizing quality loss[Griffin and Olano, 2015; Lavoué et al., 2018]. However, finding a good set of parameters for these operations is a non-trivial task. The reason for this is twofold. First, the visibility of artifacts in one texture (e.g., albedo texture) is affected by the information in other textures (e.g., normal texture), mesh geometry, illumination, and viewing direction. Therefore, all the information should be taken into account. Second, there is significant variability in platforms to which renderings are ported. The differences include display size, resolution, and brightness, but also memory and performance limits [Epic Games, 2018]. Therefore, it is critical to facilitate the process of compressing and resampling textures using an automatic process. This goal can be achieved using image metrics which are used to predict the visibility of distortion for different parameters sets. Unfortunately, as demonstrated in this as

well as previous works [Griffin and Olano, 2015; Lavoué et al., 2018], the performance of existing metrics is suboptimal with a significant room for improvements.

In this chapter, we consider the problem of automatic prediction of an optimal texture resolution for storing albedo and normal information during real-time rendering. The key challenge we address is designing a reliable and precise image metric for detecting visible distortions. While several general-purpose metrics have been proposed [Mantiuk et al., 2011c; Wolski et al., 2018], we demonstrate that their prediction can be significantly improved when they are trained on a task-specific dataset. The problem, however, is that retraining such metrics from scratch involves collecting a massive amount of experimental data from labor-intensive user experiments such as marking experiments [Wolski et al., 2018]. To address this issue, we demonstrate that one can retune existing metrics using the data from much simpler and shorter user experiments. The main contribution of this chapter is a strategy for retuning the general-purpose CNN-based visibility metric proposed in Chapter 3 (denoted in this chapter as CNN-GP) with a new dataset collected in a modest-in-scope perceptual experiment. Using this data, CNN-GP can be turned into a reliable metric for selecting texture resolution, with a significantly higher accuracy of prediction for this given task. While this chapter demonstrates the performance of such a procedure in the context of texture optimization for game engines, our experiment design and retuning can be potentially used also in different applications.

4.2 Overview

The goal of this chapter is to provide a technique for choosing texture resolution used for rendering complex game environments. We cannot directly use a state-of-the-art visibility metric, such as the CNN metric proposed in Chapter 3 (referred as CNN-GP) or HDR-VDP-2 [Mantiuk et al., 2011c], as they are designed under the assumption that the observer takes an arbitrary amount of time to attend each region of an image. We observed that this assumption is too conservative for our target application of a video game, in which a player immersed in a virtual environment is less likely to spot every small difference.

A possible solution to the above problem is to retrain the metric using a new application-specific dataset. However, collecting new experimental data with local marking (which is required for a visibility metric) for many objects, textures, and viewing position is impractical due to the amount of time needed for subjective studies. Moreover, it is unclear how to design a marking experiment that would reflect limited observation time expected in a game-like scenario.

In this work, we demonstrate that it is possible to retune existing CNN-GP metric using a much smaller, application-specific dataset, collected using a more efficient experimental procedure. To this end, we collect a new dataset[1] in Experiment 1, in which the observers are asked to determine the smallest texture size for which quality degradation is not visible (Section 4.4). We also collect a smaller dataset of local visibility maps in Experiment 2 (Section 4.5), which we use only for validation. Since the dataset from Experiment 1 provides only a single detection probability per image, we cannot use it directly to retune the CNN-GP metric, which requires local information. Our key idea to overcome this problem is to exploit existing visibility metrics, such as Butteraugli, HDR-VDP-2, or CNN-GP, and use them to produce dense visibility maps. While these maps are inaccurate, we demonstrate that it is possible to adjust them using our new dataset. To do that, we first find a pooling function

[1] https://doi.org/10.17863/CAM.43380

(Section 4.6), which relates the predicted visibility maps to the collected subjective data. Then, we generate a large set of visibility maps using three existing visibility metrics and adjust them using the data collected in Experiment 1 (Section 4.7). Given the large number of locally labelled images, we can much improve the prediction performance for our dataset (Section 4.8) and demonstrate good performance in our application (Section 4.9). The schematic diagram of our pipeline is illustrated in Figure 4.1.

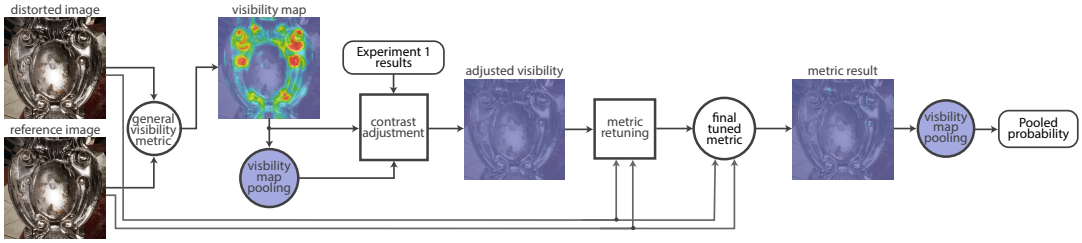


FIGURE 4.1: *The procedure for retuning a visibility metric. The reference and distorted images are used to generate the distortion visibility maps by one of existing VMs. Then, the thresholds from an application-specific perceptual experiment and pooled probabilities are used to adjust the visibility in the map. The resulted data is used for retuning a CNN-based metric that computes application-specific visibility maps. To asses the optimal texture resolution for target application the resulting visibility map is pooled to a single artifact detection probability value using same pooling strategy as one used for visibility adjustment.*

4.3 Background

This section explores the specialized background and related work on the application side of this chapter. We discuss solutions conceptually similar to our search for optimal texture resolution, focusing mainly on the level of detail (LoD) for textured 3D models and texture compression masking.

4.3.1 Level of Detail for Textured 3D Models

Controlling *level-of-detail (LoD)* for textured objects poses similar challenges as our application. Mesh simplification introduces complex interactions between texture and geometry, which influence the visibility of distortions [Lavoué and Mantiuk, 2015]. Initial solutions relied on simple geometric error measures such as *texture deviation*, which accounts for a given texel motion in the object space as a result of mesh simplification [Cohen et al., 1998]. Following a general observation that image-space metrics often lead to more robust quality predictions [Lindstrom and Turk, 2000], Tian and AlRegib [2004] proposed an image-space approximation of texture deviation.

The visual masking of geometry details by texture has been widely investigated [Ferwerda et al., 1997a; Qu and Meyer, 2008; Holly E. Rushmeier, 2000], and complex perceptual image-space models inspired by VDM and VDP have been proposed. Walter et al. consider using rendered textures as a masker and create a predictor based on the quantization matrices found in JPEG image compression [Walter et al., 2002]. Computationally efficient solutions have been proposed, where image-space visual masking processing was moved to vertex space [Menzel and Guthe, 2010], or just a simple contrast sensitivity function (CSF) was considered [Williams et al., 2003].

Early data-driven approaches relied on perceptual experiments that systematically measured the impact of mesh simplification and texture resolution reduction on the

perceived quality of 3D models. A quantitative model that predicts the MOS data that were obtained this way has been proposed in work by Pan et al. [2005]. Guo et al. [2016] used slowly moving animation in a 2-alternative-forced-choice (2AFC) experiment to account for multiple views. Based on the collected data, they propose a quality metric that linearly combines object geometry (e.g., the standard deviation of curvature differences) and image-space (MS-SSIM) components to account for geometric and texture distortions.

As found by Larkin and O'Sullivan [2011] silhouette simplification is the dominant artifact that should drive mesh simplification. In this work, we assume that mesh simplification is properly set up, and we focus on texture and shading artifacts in complex scenes, while still considering the geometry impact on the visibility of such artifacts.

4.3.2 Texture Compression Masking

As observed in previous work [Griffin and Olano, 2015; Lavoué et al., 2018], texture compression is essential to keep the description of large scenes in the GPU memory. Multiple textures describing surface reflectance (diffuse, gloss, and specular components) and geometry (normal vectors for normal mapping effects) are required to achieve high-quality shading. Such textures interact not only with mesh geometry, as discussed in the previous section, but also mutually mask each other as a function of lighting and view directions.

Griffin and Olano [2015] investigated this effect and advocated for evaluating texture compression directly in the scene environment for stochastically sampled viewpoints. They observed significant differences in CIELAB ΔE_{94} and SSIM metric predictions with varying viewpoints, texture type and content, which precludes a unified compression setup for all textures. They concluded that current GPU compression algorithms are too conservative, and the bitrates can be significantly reduced while maintaining the rendering quality.

Lavoué et al. [2018] performed a series of perceptual experiments to investigate the masking problem between diffuse and normal textures. They ran a 2AFC experiment to derive the compression visibility threshold with respect to non-compressed reference textures. They used planar texture patches with a fixed 128×128 resolution and 5 different compression levels as stimuli. They reported a limited performance of existing metrics such as SSIM, MS-SSIM, HDR-VDP2, and PSNR in predicting the perceptual data, in particular for the normal texture compression. They compared the SSIM prediction for illuminated complex geometry and planar patches (as in their experiment), and obtained a mean correlation of 0.59, which is significantly better than a naive comparison of 2D compressed textures where only 0.47 correlation was achieved. However, it can be expected that the actual correlation between the SSIM prediction for the planar patches and the human subject judgment for complex geometry and lighting can be even lower.

In this chapter, we draw inspiration from these previous works. We follow the approach of Griffin et al. and use rendered images to obtain a robust measure of artifact visibility. Our rigorous 4AFC threshold measurement procedure is similar to the one used by Lavoué et al. However, we extend beyond those works by proposing a task-specific visibility metric for texture resolution selection. We achieve this goal by retuning an existing CNN-based metric and propose an inexpensive way of collecting perceptual data that are required for this task.

4.4 Experiment 1: Detection Thresholds

The purpose of our first experiment was to collect data regarding the minimum texture resolution that does not introduce visible artifacts in rendered objects.

Stimuli We chose 33 complex 3D models from the Unreal Engine Marketplace. They include characters, creatures, buildings, and objects. Each of them has detailed geometry, high-resolution textures, and uses complex materials. The objects were placed in different environments that are shipped with Unreal Engine, which allowed us to analyze the objects' appearance under different illumination conditions: directional light, point light, spotlight, skylight, and precomputed light probe. For each object, we considered one or two unique camera positions to capture texture details and shading variations. We rendered the scenes using the Unreal Engine 4 [Karis, 2013]. The target objects were outlined with a red stroke to prevent them from blending with the background.

We created stimuli for the experiment by rendering the scenes with different resolutions of albedo and normal maps. We started with a resolution of 1024×1024 pixels for each texture, and we decreased it in 20-pixel steps down to 24×24 pixels. To limit the number of stimuli, we considered five different downscaling scenarios: reducing the resolution of (1) the albedo map only, (2) the normal map only, (3) both the normal and the albedo maps by the same amount, (4) the normal map twice as much as the albedo map, and (5) the albedo map twice as much as the normal map. The downscaling was performed using Lanczos filter. Using the above procedure, we generated 127 scenes with 51 combinations of resolutions for albedo and normal maps, which resulted in 6477 images. An example of stimuli is presented in Figure 4.2.

For rendering, we did not consider texture compression as it was done in work by Lavoué et al. [2018]. Instead, we used a non-compressed RGB format due to Unreal Engine 4 design philosophy to refrain from texture compression until final packaging. This philosophy is motivated by the fact that each platform has different requirements regarding CPU, GPU, memory, and texture format. Another concern is that a mipmap level alone could determine the optimum texture resolution. For example, if the rendered image refers at most to the mipmap level 1 and never refers to mipmap level 0 (the highest resolution), we know that the resolution could be reduced by half. For that reason we selected the maximum texture resolution so that mipmap level 0 is used for the most of the rendered pixels. After ensuring this, we disabled mipmapping as UE4 does not support mipmapping for non-power-of-two textures.



FIGURE 4.2: *Example of one scene from the dataset with 3 texture resolution variants of albedo map: highest resolution (left), medium resolution (middle), and lowest resolution (right). The target object in the scene has a red outline to prevent it from blending with the background.*

Experimental Procedure Each session of our experiment was split into two phases: a method-of-adjustment (MoA) phase, giving an initial estimate of the detection threshold, and a four-alternative-forced-choice (4AFC) phase, which refined the initial estimates. In the MoA phase, the observer's task was to adjust the texture resolution with the up/down arrow keys, so that the difference with respect to the reference is barely visible. A screenshot from that phase is shown in Figure 4.3. When switching between two texture resolutions, a short (0.5 second) blank screen with the color of the background was shown so that the observers could not use temporal changes to locate the artifacts.

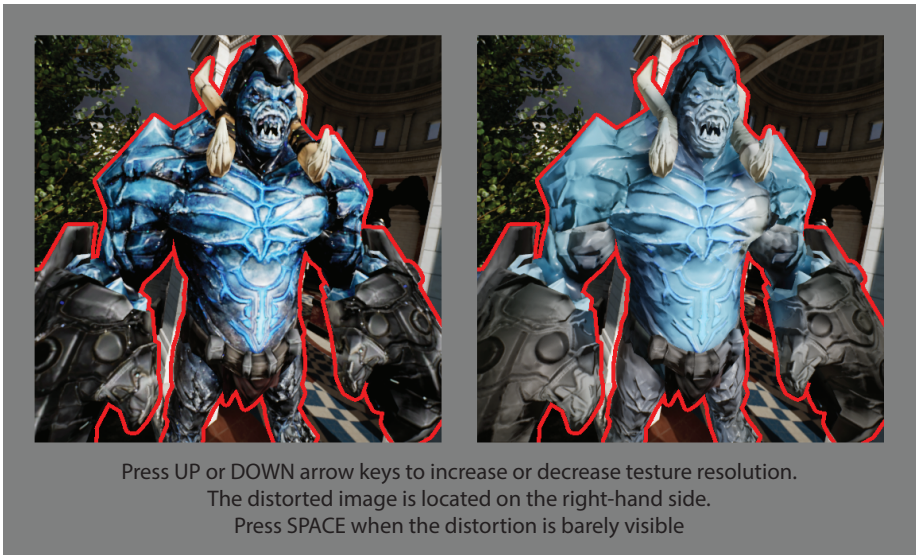


FIGURE 4.3: *Screenshot from the method-of-adjustment (MoA) phase of the experiment. On the left is the reference image rendered with the highest texture resolution, and on the right is the distorted image with reduced texture resolution.*

The second 4AFC phase of the experiment was a modified QUEST procedure [Watson and Pelli, 1983]. Four images were presented to the observer, where only one image was distorted. The position of the distorted images was randomized. The observer was asked to choose the image that was different from the others. We used 4AFC as it needs fewer trials to converge than 2AFC because of the lower guess rate, while it gives similar results [Vancleef et al., 2018]. The QUEST adaptive method was initialized with the threshold estimate collected using the MoA method in the first phase. The QUEST trials for different scenes were interleaved so that each trial typically showed a different scene. This was done to prevent the observer from learning the location of distortions. Between 25 and 30 4AFC trials were collected from each observer for each scene. The presentation time of each trial was limited to 10 seconds to ensure that the data reflects a more realistic scenario of viewing rendered models in a game environment. The screenshot from the second phase is shown in Figure 4.4.

Viewing Conditions The experiment room was darkened and the monitor was positioned to minimize screen reflections. The observers sat 96 cm from a 24″, 3840×2160 resolution Eizo CH246-4K display, resulting in an angular resolution of 120 pixels per visual degree. As such resolution is much higher than for a standard monitor seen from a typical viewing distance (45–60 pixels per visual degree), we enlarged the images to twice their original size using the nearest neighbor filter. The measured



FIGURE 4.4: *Screenshot from the 4AFC phase of the experiment. The observer's task is to choose the image that is different from the others.*

peak luminance of the display was 120 cd/m^2 and the black level was 0.17 cd/m^2. The display was calibrated to the sRGB color space using its built-in self-calibration functionality.

Observers To reduce the effect of fatigue, the experiment was split into 9 sessions, where each session lasted less than one hour. Each participant completed a subset of these sessions so that the threshold for each scene was measured by 10 participants. All observers (aged 21 to 27) were computer science students or researchers. The observers were paid for their participation. All observers had normal or corrected-to-normal vision and were naive to the purpose of the experiment.

4.4.1 Cross-Population Psychometric Function

To use the data collected in the experiment for training, we need to estimate a cross-population psychometric function: the function that explains what portion of the population can see the difference at a particular texture resolution for a given scene, as shown in Figure 4.5. To find such a function, we first fitted a psychometric function per each observer to the QUEST trial responses and estimated a 75%-probability threshold. Then, the outliers were removed by eliminating per-observer thresholds

that were more than two standard deviations away from the mean. Assuming that the threshold texture resolution is distributed normally in a population, we can model the cross-population probability of detection P_{det} as a (negated) cumulative normal distribution:

$$P_{det}(r) = 1 - \Theta(r; \mu, \sigma) = 1 - \frac{1}{2}\left[1 + \text{erf}\left(\frac{r-\mu}{\sigma\sqrt{2}}\right)\right] \tag{4.1}$$

where Θ is the cumulative normal distribution, r is the texture resolution in pixels, erf() is the Gauss error function, and μ, σ are the distribution parameters. The parameters can be found by computing the mean and standard deviation of per-observer thresholds.

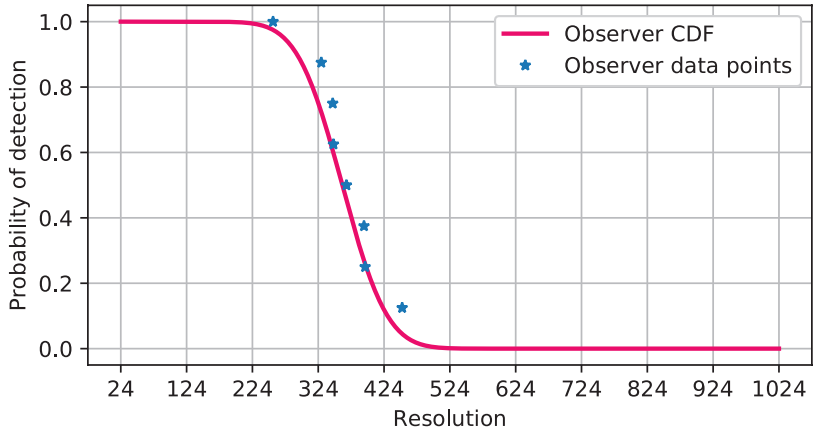


FIGURE 4.5: *Cross-population psychometric function explaining what portion of the population can see the difference between the reference image with full-resolution textures and a test image in which the texture resolution was reduced.*

4.5 Experiment 2: Local Visibility Maps

The goal of the second experiment was to collect a validation dataset with local visibility maps. We use this dataset to show the differences between both experimental protocols, but also to test our retuning procedure. We used the same experimental protocol as in Section 3.3, in which observers painted visible difference in images. Because the protocol is time-consuming, this experiment was feasible for only a subset of 32 from over 6477 images used in Experiment 1. The monitor and viewing conditions were identical to those in Experiment 1.

Experimental Procedure Observer were presented with distorted and reference images side-by-side. They were then asked to carefully mark all the areas where artifacts are visible in the distorted image using a brush of adjustable size. The final visibility maps were obtained by averaging per-observer maps.

Observers 10 participants were recruited from the same demographic group as for Experiment 1. Most of the observers were new participants and only a few observers completed both experiments.

4.6 Pooling for Task-Specific Visibility Threshold

Our goal is to determine the texture resolution for which less than 20% of observers are going to notice any artifacts. One obvious approach is to run a visibility metric, such as CNN-GP or HDR-VDP-2, at every resolution level and use the predictions to determine the desired detection threshold. But, since each of these metrics predicts local visibility maps, where each pixel has associated probability of detection, we need to pool the values from the map into a single value that corresponds to the desired threshold.

For a conservative threshold, we could use the maximum value from the visibility map, as done in the previous chapter and work by [Alakuijala et al., 2017]. Such approach, however, suffers from two problems. Firstly, it assumes that every observer will always spot the most visible distortion. While this assumption is valid for a well-controlled marking experiment (Experiment 2), in which observers scrutinize every part of an image, it does not hold for the detection experiment (Experiment 1), or the target application of texture rendering. In less controlled conditions, different observers are going to spot different distortions, and only few will spot the one that is the most visible. This is illustrated in Figure 4.6 (left), where we plot the detection thresholds from Experiment 1, vs. the maximum value from the local visibility maps from Experiment 2. The plot on the right in the same figure shows that discarding the top 5% of the values (95th percentile) when pooling improves the relation between both datasets. Secondly, the maximum value is strongly affected by the prediction error. If we were to compute an average value across the visibility map, we could expect that a portion of the prediction error would cancel out. However, if we rely on a single maximum value, we cannot take advantage of averaging multiple predicted values.

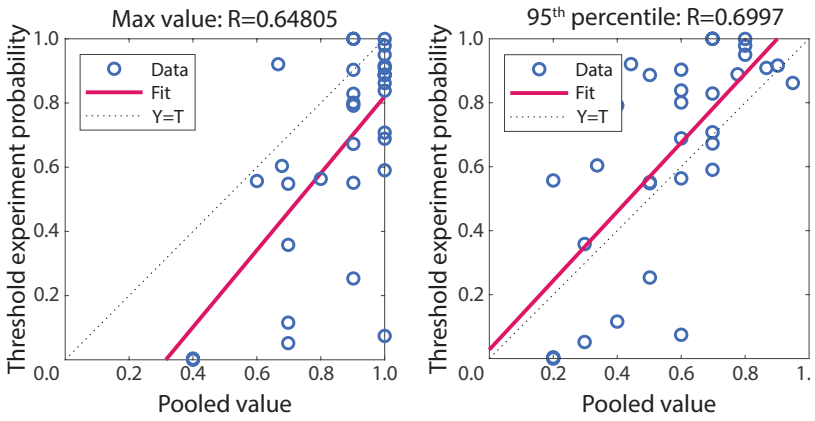


FIGURE 4.6: *The relation between the measured thresholds from Experiments 1 and 2. The scatter plots show the relationship between the maximum value (left) and 95th percentile (right) computed from the local visibility maps (Experiment 2) and the detection thresholds found in Experiment 1.*

To find a task-specific visibility threshold, we tested a number of pooling strategies on the predictions of three state-of-the-art visibility metrics: CNN-GP (Section 3.6), HDR-VDP-2 [Mantiuk et al., 2011c] and Butteraugli [Alakuijala et al., 2017]. The goal is to find a pooling that provides the lowest prediction error for the thresholds determined in Experiment 1. Firstly, we tested ten percentiles, from 90th to 99th. Secondly, we used a trainable pooling function realized by a neural network as shown in Figure 4.7. The network was composed of four convolutional layers interspersed with max-pooling layers and terminates with three fully connected layers, the last of

which produces a scalar. To prevent over-fitting, we used 5-fold cross-validation with an 80:20 split between training and testing data.

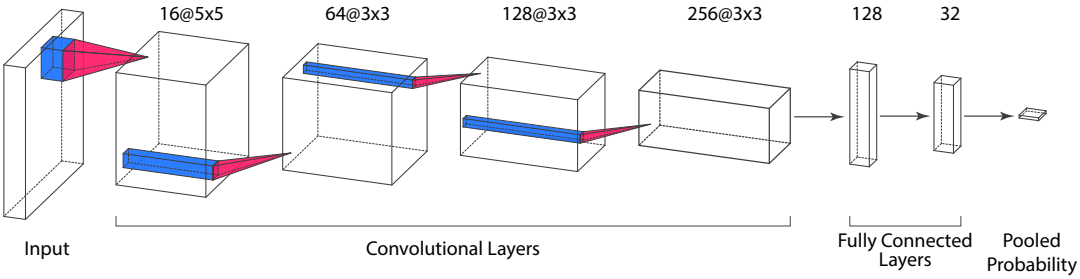


FIGURE 4.7: *The architecture of the CNN used for predicting the pooled threshold. Each convolution layer shows the number of kernels and the size of each one. The final fully connected layers show the number of neurons.*

An optimal pooling strategy is determined for each of the three metrics as follows. We run a given metric for each scene on all possible texture resolutions and consider all pooling strategies to find the probability of detection. To regularize the predictions across the resolution levels, we fit for each pooling strategy a psychometric function from Equation 4.1. Then, we compute a mean-absolute-error (MAE) between texture threshold resolution found in Experiment 1, and the texture resolution corresponding to the assumed 20% of probability of artifact detection derived from the psychometric function for this pooling option. We repeat this procedure for all scenes. The lowest prediction error was achieved for 93^{rd} percentile for CNN-GP and Butteraugli, and 94^{th} percentile for HDR-VDP-2. We were not able to train NN to provide higher correlation than these percentiles. Figure 4.8 gives an example of a single scene for which the psychometric functions were fitted for all the tested metrics. In Figure 4.8 we also present the maximum error value as a pooling option, which we denote as CNN-GP-Max. When comparing the match of CNN-GP-Max and CNN-GP-P93 to the experimental results, the benefits our pooling optimization are clearly visible.

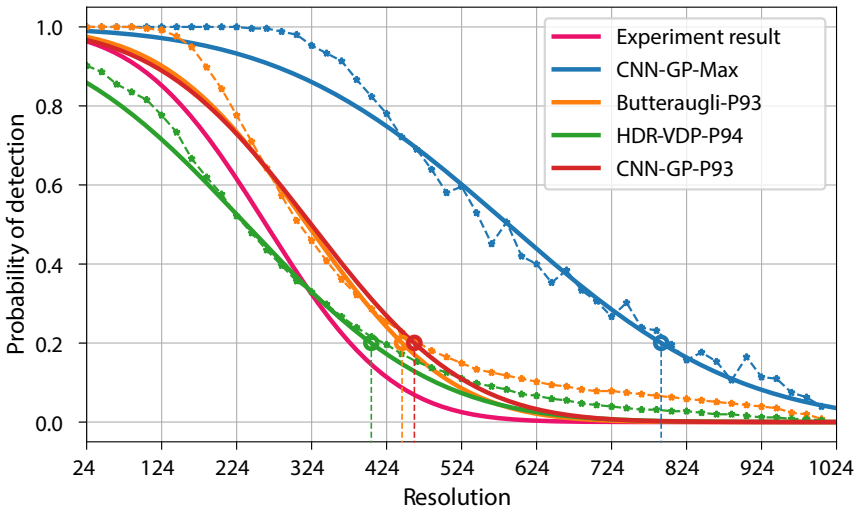


FIGURE 4.8: *Metric predictions (dashed lines, only for CNN-GP-Max, HDR-VDP-P94, and Butteraugli-P93) and psychometric functions fits (continuous lines) for a chosen scene. The desired resolution is selected for the probability of detection 0.2.*

Figure 4.9 compares optimal percentile pooling (continuous lines) to the results achieved with max-value pooling (dashed lines). The results are shown as an error histogram over all available scenes. The plots show that the trainable pooling also

reduced the prediction error as compared to naïve max-value pooling, but provided worse results than the best percentile strategy.

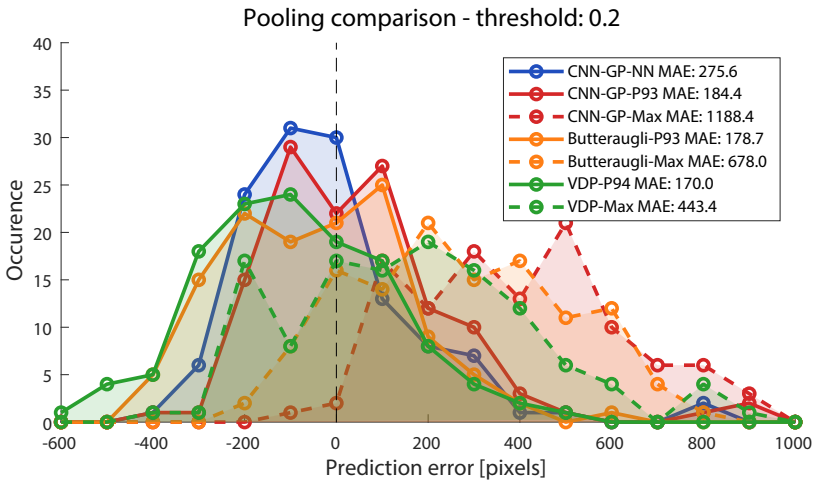


FIGURE 4.9: *Histograms of the error distribution for pooled probability value for scenes considered in Experiment 1. The error is expressed as signed texture resolution (both horizontal and vertical) differences in pixel units with respect to the experimental observer data. The percentile pooling (continuous lines) results in much lower maximum absolute error (MAE) values than naïve max value pooling (dashed lines). Trainable pooling strategy (NN Pooling) also performs reasonably well.*

4.7 Metric Retuning

Although the percentile pooling much improves the prediction performance, it is too simple to account for complex changes in visibility thresholds in our application of texture rendering. We have also shown that complex NN-based pooling generally does not provide better performance. To further improve the metric performance, we need to retune it to a new application. As the CNN-GP provides the most flexibility of retraining, we focus on this metric.

4.7.1 Retuning Dataset

Before CNN-GP can be retuned for our texture optimization application, we need to generate a dataset consisting of pairs of reference and distorted images and the corresponding per-pixel visibility maps. Having such maps is vital for training a CNN-based metric, which operates on patches. Since our experiment does not provide per-pixel visibility information, we use instead existing visibility metrics, CNN-GP itself, HDR-VDP-2 and Butteraugli, to generate such maps. Each of these metrics was previously re-trained on the LocVis dataset (Section 3.2) as described in Section 3.5, which made them more accurate and robust to a broad set of artifact types. As the visibility maps predicted by existing metrics are not suitable for the target application because of the reasons outlined in Section 4.2, they need to be adjusted using the newly collected experimental data. The overview of the retuning procedure is illustrated in Figure 4.1.

First, we need to understand how visibility maps are produced by each visibility metric. Chosen visibility metrics involve several stages of spatial processing and point-wise nonlinearities, which culminate in transforming their semi-final output, a perceptually-normalized difference of contrast C, into probabilities using a sigmoidal

psychometric function ψ:

$$\tilde{P}_{det}(x,y) = \psi(C(x,y)) \,, \tag{4.2}$$

where x and y are pixel coordinates. The per-image probability of detection is computed as the the optimal percentile value (Section 4.6) of $\tilde{P}_{det}$ over the image. CNN-GP uses a standard, non-parametrized sigmoid function:

$$\psi_{CNN}(c) = \frac{1}{1+e^{-c}} \tag{4.3}$$

In HDR-VDP-2 and Butteraugli the psychometric function has the form:

$$\psi(c) = 1 - e^{\ln(0.5)\, c^{\beta}} \,, \tag{4.4}$$

where c is the image difference (in perceptual contrast space) and β (typically 3.5) controls the slope of the psychometric function. Because the psychometric function ψ saturates at large positive and negative values and introduces the loss of information after discretization, we need to adjust the values of difference contrast C rather than probabilities $\tilde{P}_{det}$ to generate visibility maps consistent with our experiment. This means that for each series of images corresponding to the texture resolution of particular scene, we find a linear transformation of C that, when transformed by Equation 4.2 and pooled using the 93^{rd} percentile, results in the detection threshold closest to the Experiment 1 results (in terms of MSE). The best linear transform is found using exhaustive search over the range of bias values. The adjustment is illustrated in Figure 4.10 in which the original metric predictions (dashed blue line) over-predicts the probability of detection. However, after adjusting C, the predictions (solid blue line) are close to the experimental data (pink line).

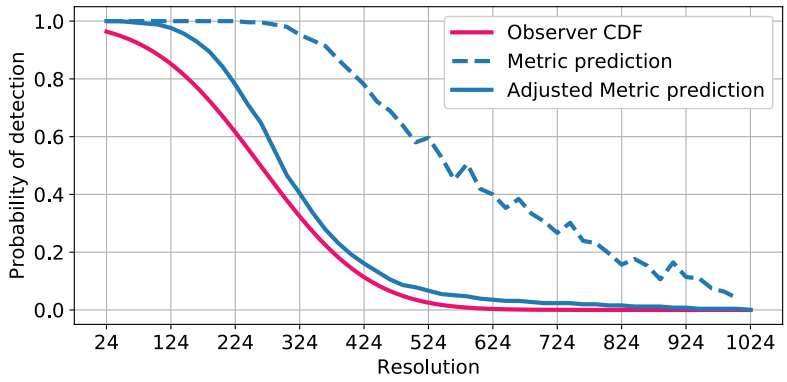


FIGURE 4.10: *Adjusted CNN-GP prediction (solid blue line) better matches observer cross-population psychometric function (pink line) than original CNN-GP prediction (dashed blue line).*

We separately generate three retuning datasets, each using a different visibility metric: CNN-GP, HDR-VDP-2, and Butteraugli. Each dataset consists of all images used in the experiment: 51 down-sampling factors, 33 3D models and all 127 variants. If the pooled value of the adjusted visibility maps differs by more than 0.05 from our experimental data, we remove that image from the dataset. This results in an extensive set of approximately 3,000 image pairs and per-pixel visibility maps, suitable for training a CNN-based metric.

4.7.2 Loss Function

In Section 3.4, we proposed a probabilistic loss function, which was designed to account for inaccuracies of manual marking but also to conservatively predicts all visible distortions, even those that are more difficult to find. However, since our dataset is automatically generated by the metrics and accounting for the difficulty of finding artifacts is essential for our application, such a loss function is unsuitable.

To match probability predictions to the generated dataset, we have experimented with Kullback-Leibler KL-divergence, but that resulted in poor predictions. Instead, we use the loss that penalizes probability differences close to the range bounds:

$$L = \sqrt{\sum_{(x,y)\in\Theta} \left[g\left(p_{label}(x,y)\right) - g(p_{pred}(x,y))\right]^2}, \tag{4.5}$$

where:

$$g(v) = \ln \frac{v+\epsilon}{1-v+\epsilon}, \tag{4.6}$$

Θ is the set of all pixels with coordinates (x, y), p_{label} is the target probability generated in the adjustment process, p_{pred} is the metric prediction and ϵ is a small constant.

4.7.3 Transfer Learning and Testing Performance

The dataset is partitioned into 5 equal parts for 5-fold cross-validation. Each 3D object/texture is present in only one fold. The network is initialized with the weights used in the original CNN-GP metric and then retrained over 50000 iterations using the new dataset. We experimented with training only the final deconvolutional layers or the entire network and achieved very similar results. We report the results for the latter.

The CNN is trained on 48×48 pixel non-overlapping patches with a batch size of 128, in Tensorflow v.1.7.0[2] with Python v3.6. We use Adam optimizer [Kingma and Ba, 2014] with the learning rate of 0.00001 and set the dropout to 0.5.

The metric is intended for offline use. Processing of one 800×600 image takes about 3,5 seconds on NVIDIA GeForce GTX 1070 and Intel Xeon E5-1650.

4.8 Results

First, we want to determine which of the existing visibility metrics produces the retuning dataset that gives the best predictions. We denote new metrics trained on the three retuning datasets as CNN-T-CNN-GP-P93, CNN-T-Butteraugli-P93, CNN-T-HDR-VDP-P94, where the label contains CNN-T (CNN reTunned), followed by the name of the existing metric used to generate the retuning dataset and the percentile pooling function. To evaluate the accuracy of retuned metrics we used the same approach as described in Section 4.6. We run each metric on all possible texture resolutions and use the optimal percentile of the visibility map to find the probability of detection for each resolution. Then, we fit a psychometric function and compute a prediction error for threshold 0.2.

The results of 5-fold cross-validation are shown as the histogram of errors in Figure 4.11. The legend includes the maximum-absolute-error (MAE), which we use instead of RMSE as it is more robust to outliers. The top left plot shows that

[2]https://www.tensorflow.org/

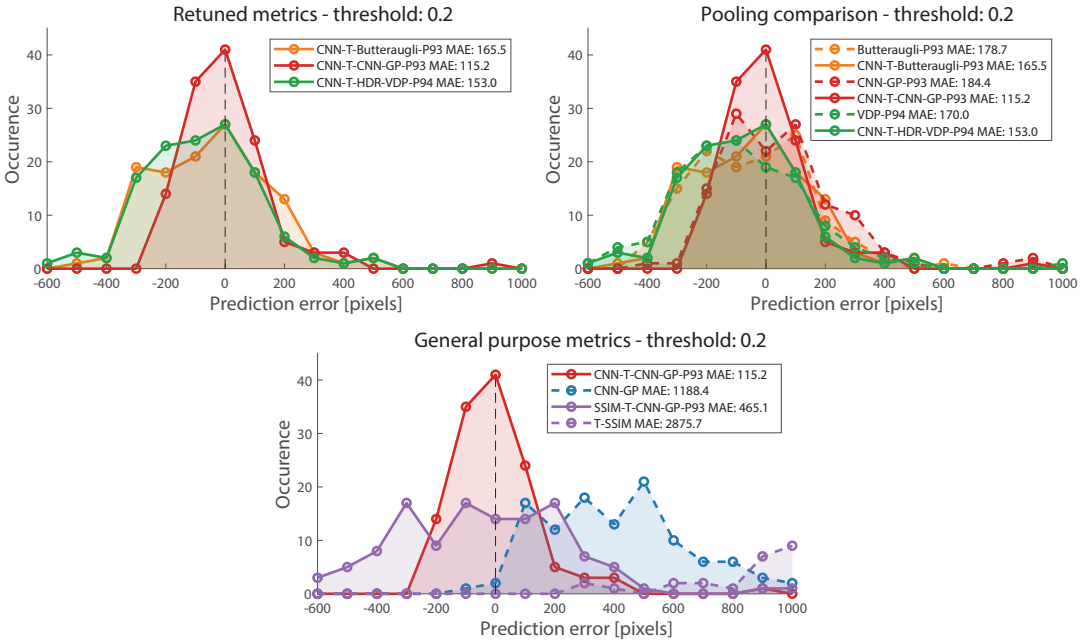


FIGURE 4.11: *The error distribution for the retuned and existing metrics. Top left: comparison of three retuning datasets. Top right: the performance gain of retuning as compared to pooling. Bottom: comparison with the existing metrics. The existing visibility metrics, CNN-GP and T-SSIM, predict the threshold for higher than required texture resolution (bottom). The prediction error in histograms is expressed as signed texture resolution differences (both horizontal and vertical) in pixel units with respect to the experimental data. The value in the legend denotes mean-absolute-error (MAE).*

retuned CNN-T-CNN-GP-P93 results in a lower prediction error than retuned CNN-T-Butteraugli-P93 and CNN-T-HDR-VDP-P94. The distribution for the latter ones is also skewed towards predicting lower resolution.

Second, we show the improvement obtained by retuning the metrics as compared to using existing metrics with the percentile pooling. The top right plot of Figure 4.11 shows that the prediction error of CNN-GP was reduced from MAE=184.4 to MAE=115.2 after retuning, demonstrating that retuning is an effective mean of improving metric predictions. The plot also shows that CNN-GP retuned using a dataset generated with HDR-VDP-2 or Butteraugli performs better than the original metrics.

Finally, we compare our best performing metric (CNN-T-CNN-GP-P93) with the predictions of the existing visibility metrics: CNN-GP and SSIM, which was found to be a good indicator of texture distortions [Lavoué et al., 2018]. We use T-SSIM version, which was augmented with the psychometric function in Chapter 3 to predict visibility rather than image quality. For a fair comparison, we also retrained T-SSIM on the same dataset as CNN-T-CNN-GP-P93. We denote such a retuned SSIM as SSIM-T-CNN-GP-P93. The bottom plot of Figure 4.11 shows very substantial improvement obtained by our retuning procedure, as compared to the existing visibility metrics. The predictions of retuned SSIM-T-CNN-GP-P93 are much improved as compared to T-SSIM, but still much worse than those of the retuned CNN-T-CNN-GP-P93.

4.8.1 Visibility Map Improvement

Despite the fact that our target application requires only a single pooled value, we can also observe improvement in the visibility map predictions. To evaluate visibility maps predictions, we compare them to 32 manually marked visibility maps from

TABLE 4.1: *All indicators show that retuned metrics provide visibility with higher accuracy than general-purpose metric CNN-GP.*

Image Metric	r	ρ	RMSE	Loss
CNN-GP	0.706	0.789	0.214	0.311
CNN-T-CNN-GP-P93	0.792	0.832	0.179	0.272
CNN-T-Butteraugli-P93	0.663	0.769	0.2	0.285
CNN-T-HDR-VDP-P94	0.715	0.791	0.167	0.248

Experiment 2. Table 4.1 shows that the retrained metrics improve predictions over the vanilla CNN-GP metric. We compared maps using the following indicators: Pearson correlation coefficient (r), Spearman correlation coefficient (ρ), Root Mean Squared Error (RMSE), and the loss function described in Section 4.7.2. Additionally, Figure 4.12 presents examples of improved visibility maps produced by the retuned metric CNN-T-HDR-VDP-P94.

4.9 Texture Size Optimization

To test performance in reducing the texture resolution without introducing visible artifacts, we generated a new set of 3D models. Four of them: *TreasureBox*, *Plant*, *DemonDoor*, and *AngelStatue* are presented in Figure 4.13. Initially, all of them are textured with albedo and normal maps at the original texture resolution of 1024×1024. We placed the models in different environments, which provided realistic lighting.

To determine the minimum texture resolutions, we first sample viewing and illumination directions from those that can be encountered in a game scenario. In Figure 4.14, we show 5 out of 8 such views. Next, each view is rendered using all possible combinations of normal and albedo map resolutions. Here, we restrict our choice to resolutions equal to the powers-of-two, which are more relevant for the game-engine application. All scene renderings are then compared to the reference rendering with full-resolution textures using the proposed visibility metric (CNN-T-CNN-GP-P93). The outcome of this prediction is summarized in the tables shown in Figure 4.14, which contain the probability of spotting artifacts for all combinations of normal and albedo map resolutions. Given these matrices, we can choose the combination which results in the smallest memory footprint and, at the same time, meets our visibility criterion. One possible example of such a criterion could be that at most 10% of the sampled views have a probability of detection greater than 0.2. Table 4.2 summarizes the optimal texture size selection and the resulting reduction in memory footprint. It shows that the optimal texture resolution varies substantially with the model, making it hard to choose the suitable texture configuration manually. The proposed metric can greatly facilitate this task.

Figure 4.13 shows the final rendering of two scenes using optimized texture size compared to reference textures and sub-optimal texture size choice. In *Plant* scene, leaf veins are the most important fine details. The image rendered with optimal texture size preserves these details, while for the rendering with the sub-optimal texture resolution, they tend to get blurred. In the second example, *TreasureBox*, the emblem on the chest's lid gets darker due to the downsampling of albedo texture. The artifact is not present in the optimal-resolution image. In *AngelStatue* scene, a high-frequency stone is present on the surface. For optimal texture resolution, the pattern remains preserved. It is not true for the sub-optimal texture resolution. In the last scene *DemonDoor*, we can observe cracks and scratches on the horns. Once again,

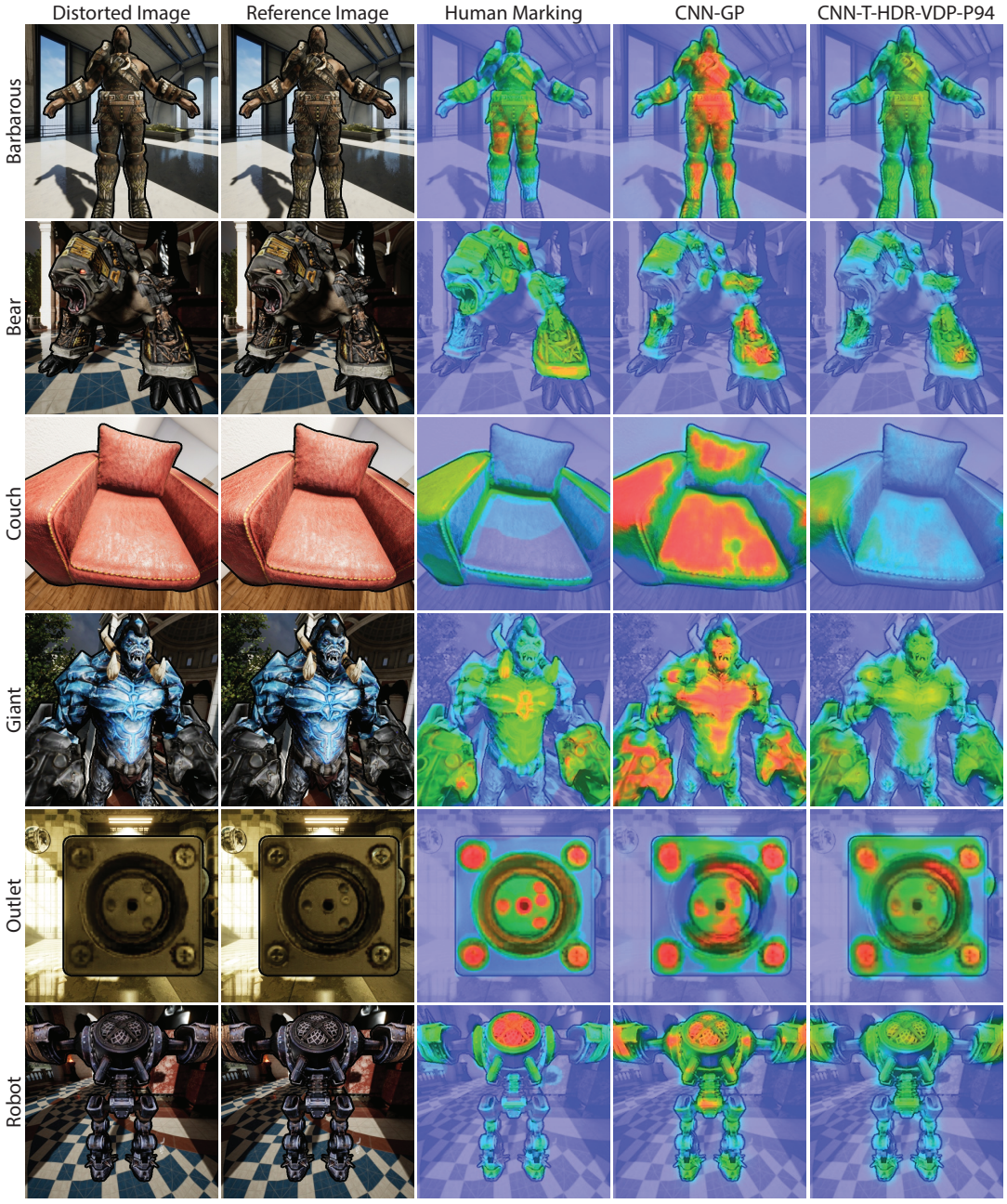


FIGURE 4.12: *Examples of visibility map alteration. In the first four rows, the retuned metric (CNN-T-HDR-VDP-P94) reduced the probability in the correct way, providing visibility maps much closer to the human marking than the general-purpose metric (CNN-GP). In the second last row, the retuned metric reduced the probability in the incorrect way (circular dents in the center of the image). The last row presents correct probability reduction (arms and body of the robot) and incorrect reduction (head of the robot).*

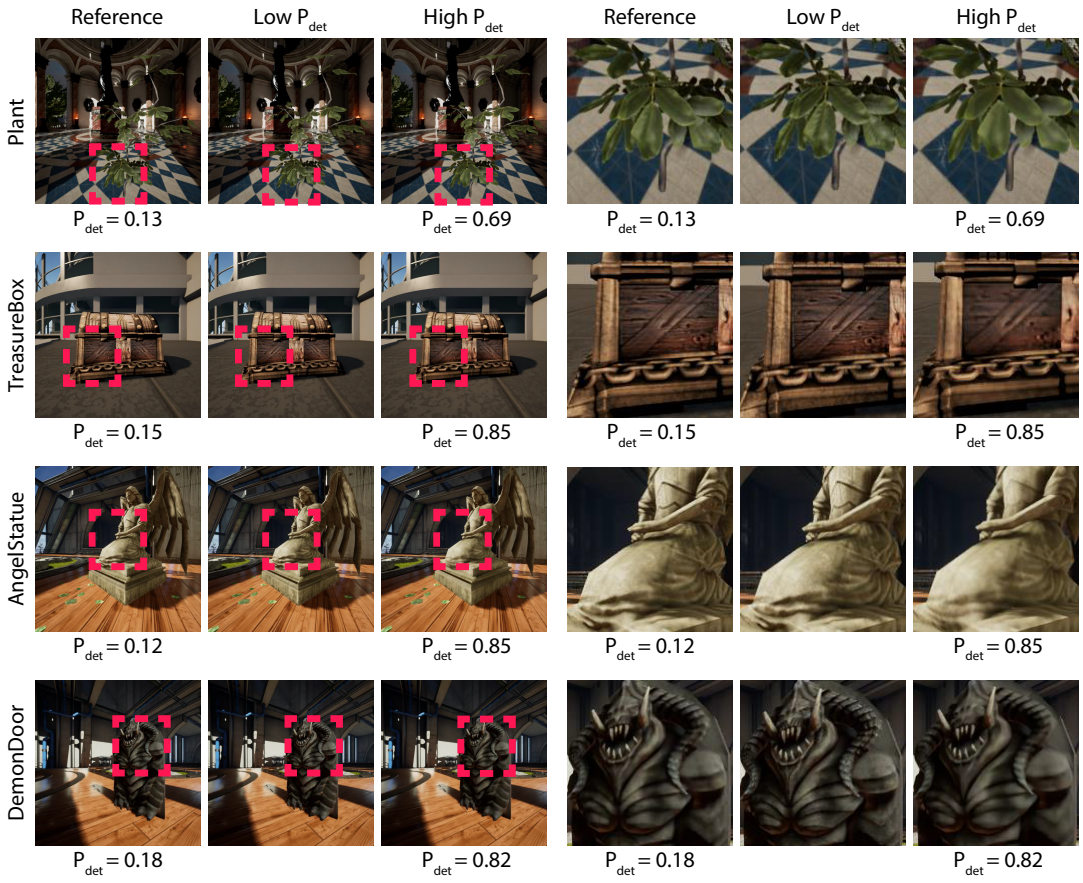


FIGURE 4.13: *Examples of texture size optimization. From left in each row: rendering with a low probability of detecting the artifacts, rendering with full texture resolutions, and with a high probability of detection. The following three images in the row detail respectively the regions in the boxes. For proper viewing conditions at the 60 cm viewing distance, the images and the insets should span 9 cm and 3 cm on the screen, respectively.*

the details are correctly preserved for optimized textures while they disappear when the sub-optimal textures are used.

The results in Figure 4.14 demonstrate a general trend where normal maps can be reduced to lower resolutions than albedo maps. We can also notice an interaction between the textures. For example, for the *Plant* model, the lower resolution of the albedo map results in strong artifact visibility with a relatively weak impact of the normal texture resolution. In *TreasureBox* for most of the camera positions, both the albedo and normal texture resolutions are important, and only moderate resolution reduction is possible (also refer to Table 4.2).

TABLE 4.2: *The texture size reductions guided by the proposed metric and the corresponding memory savings. The resolution reported for the scenes shown in Figure 4.14.*

Scene	Albedo [px]		Normal [px]		Memory [MB]	
	before	after	before	after	before	after
TreasureBox	1024	512	1024	512	8	2
Plant	1024	512	1024	128	8	1.1
DemonDoor	1024	256	1024	128	8	0.3
AngelStatue	1024	512	1024	512	8	2
King	1024	1024	1024	1024	8	8

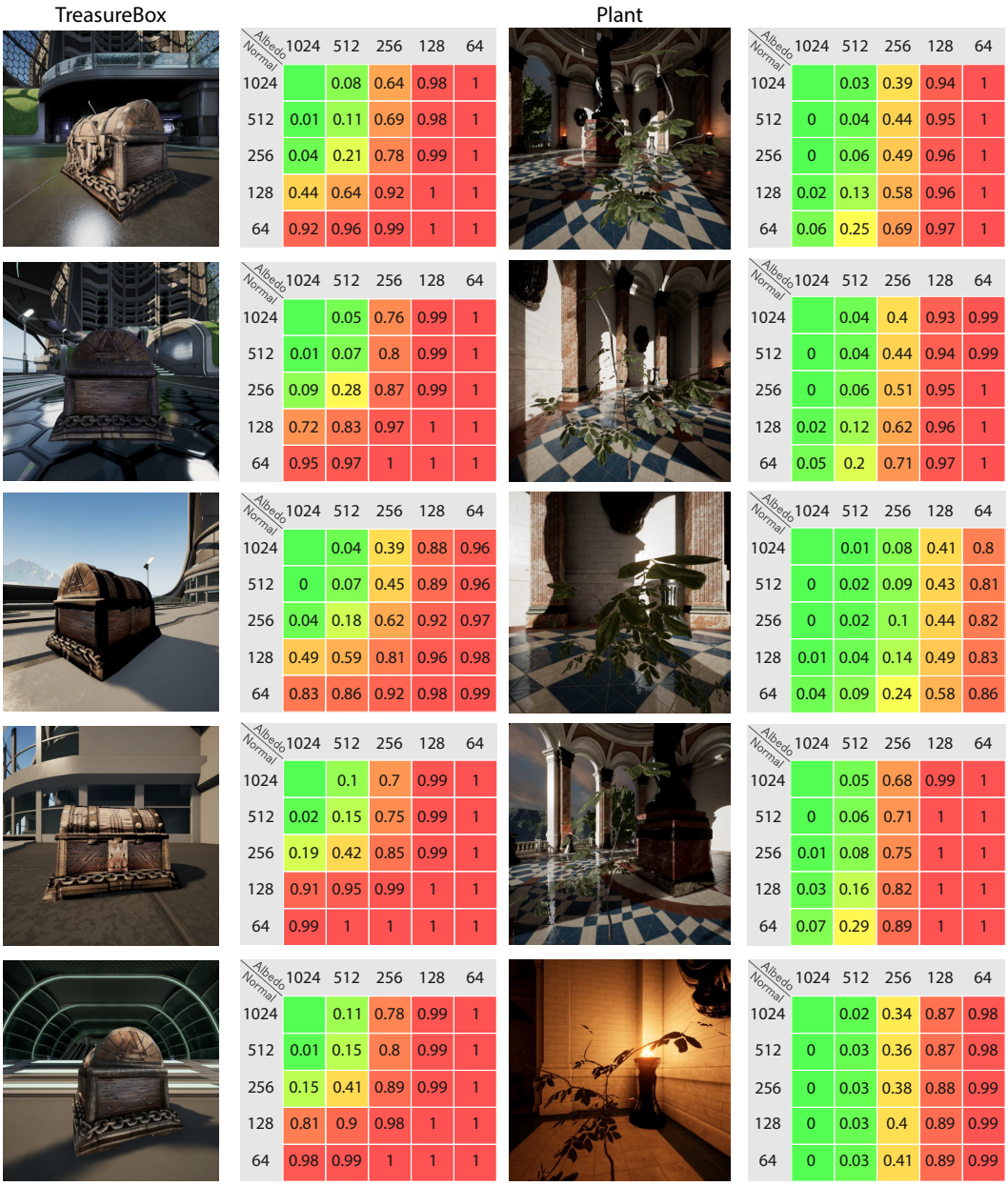


FIGURE 4.14: *Examples of 5 views rendered using reference texture (rows) and the corresponding metric predictions for all combinations of normal and albedo texture resolutions. The columns in the tables represent variations in the albedo map resolution, and the rows the variation in the normal map resolution. The color coding represents the probability of detection, from 0 (green, invisible) to 1 (red, definitely visible).*

4.10 Limitations

As any learning-based method, the proposed visibility metric is restricted to the examples that are close to the training dataset. For example, our dataset considers only albedo and normal maps, while 3D assets may also contain specularity, transparency, anisotropy, and other types of texture maps, for which our metric can be less accurate.

4.11 Conclusions

Predicting the visibility of artifacts in rendering is a challenging task. Existing state-of-the-art metrics are often trained for a specific task, such as detecting side-by-side difference, which may not be applicable in many scenarios, and therefore, lead to over-conservative predictions. We demonstrate how a CNN-based metric can be easily retuned for a specific task using a dataset collected with limited effort. Such retuning leads to a greatly improved prediction performance, which lets us use our new metric in a practical application of optimizing the texture resolution of 3D assets.

Chapter 5

Perceptual Dataset of Mesh Distortions

5.1 Introduction

Subjective quality datasets are a required component to develop models and algorithms which follow users' preferences and generate high-quality results. Creating these datasets is a challenging task, which requires a comprehensive understanding of perception, psychophysics, statistical analysis, and modeling. Although large-scale perceptual databases such as LIVE [2005], TID [2015b], or LocVis (Section 3.2) exist for image and video applications, many other areas of graphics lack this support.

In this chapter, we study the perception of geometric distortions on faces. Human characters or avatars are ubiquitous in gaming, entertainment, and social applications. Faithful reproduction of faces is a particularly important and challenging task [Zhao et al., 2003], as people are known to be especially sensitive to distortions of facial features. Although these distortions can be measured objectively, this does not provide a sense of how users perceive their impact. Quantifying this is important because perceptually-driven methods in geometry processing can result in improved outcomes for key applications by selecting optimal cost vs. quality tradeoffs.

We initiate this chapter with a discussion on existing datasets of meshes representing faces (Section 5.2). Then, in Section 5.3, we describe our effort of creating a novel dataset consisting of 100 high-quality face meshes, taking special care to maintain demographic balance. In addition, this section includes a step-by-step process for applying a meticulously curated selection of geometric distortions, as depicted in Figure 5.1. Following, we use this dataset in Section 5.4 to measure the perceived magnitudes of these artifacts in a large-scale user study, which gathered over 84,000 subjective judgments. Next, we demonstrate the insights on the perception of mesh distortions (Section 5.5) and evaluate the performance of existing geometry metrics against our ground-truth data. (Section 5.6).

In Section 5.7, we demonstrate some promising use cases for our perceptual database. We propose a simple data-driven retargeting scheme to augment these metrics and leverage the augmented metrics to perceptually tune algorithms like mesh compression and level-of-detail rendering (LOD). We conclude this chapter in Section 5.8 by mentioning gathered insights, limitations, and possible directions for future work.

Our perceptual database and benchmark dataset of reference and distorted faces are made available[1] with the goal of encouraging further research into perceptual aspects of computational geometry.

[1] https://github.com/facebookresearch/Geo-metric

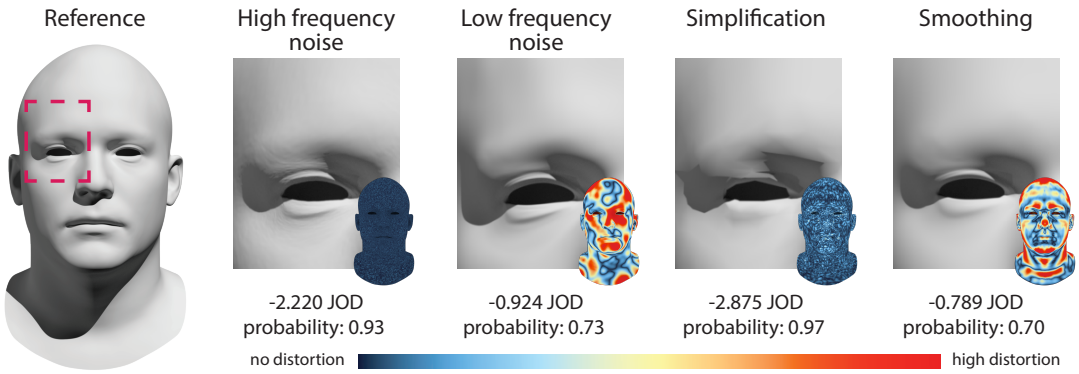


FIGURE 5.1: *A sample of the artifacts used in our subjective study. Heatmaps show the scale of each distortion, expressed as Euclidean distance to the reference. Just-objectionable-difference (JOD) values give an estimate of the perceived magnitude and can be converted to a probability of selection, shown below.*

5.2 Face Perception and Datasets

Several datasets of 3D faces have been presented in the literature, categorized as real (captured), or synthesized. The former are typically obtained using 3D scanners. Cao et al. [2014] use a Kinect to scan 150 individuals and 20 expressions. Due to the low precision of the depth estimates, the detailed facial geometry such as wrinkles is lost. Zhang and colleagues [2013] use a Di3D dynamic face-capturing system to capture a 3D video database of spontaneous facial expressions from 41 young adults. Bagdanov et al. [2011] capture 3D scans from 53 Caucasian adult individuals using the 3dMDFace system. Ranjan et al. [2018] released a database including 12 individuals with 12 extreme expression sequences. To explore lip synchronization and realistic facial animation, Cudeiro and colleagues [2019] released a 3D audio-visual dataset, captured from 12 individuals with 40 dynamic sequences for each. Zhu et al. [2021] present a large-scale detailed 3D face dataset, captured from 938 Asian participants, each with 20 specific expressions. Although these datasets provide good coverage over age and gender, they rarely take other demographic factors into consideration. Yin et al. [2006] aims to capture expressions of 100 subjects from several demographic groups, providing 2500 unique face meshes with high-quality textures and low-detail geometry. In a follow-up study [2008], the authors add time-varying data, providing 606 3D expression sequences with 100 frames each.

Unlike captured geometry, synthetic datasets are often generated using a 3D Morphable Model (3DMM) [Blanz and Vetter, 1999] using either a statistical appearance model [Paysan et al., 2009; Guo et al., 2018], FACS [Cosker et al., 2011], or inverse rendering [Booth et al., 2018]. Such datasets are often orders of magnitude larger and can cover more granular data groups. However, reconstructed faces often suffer from loss of detail, making them less accurate representations of reality. Our synthetic data is based on high-resolution scans gathered using a 3dMD scanner. In addition, our population distribution aims to reflect diverse US demographics, based on the Census [2021], which ensures some representative data is present for each group.

5.3 Synthetic Dataset

To measure the perceptual effects of distortions on 3D facial geometry, we first need to prepare a representative benchmark dataset. We do this by leveraging a large-scale private collection of human head scans. Although we are not able to use these proprietary scans in this work, we draw on this database to generate a synthetic set which blends the original scans into new meshes, while maintaining desirable properties like high quality and demographic balance.

5.3.1 Non-Public Dataset

Proprietary head meshes were gathered using a commercially available high-quality 3D scanner: 3dMD Ltd's static 12-viewpoint 3dMDhead scanning system. This is a top of the line scanner which includes 36 machine vision cameras in 12 Modular Camera Units (or viewpoints), optimized specifically for full head and ear detail capture. The system automatically generates a continuous 3D textured surface mesh with variable resolution (typically around 180k vertices). The 3dMDvultus Software is then used to add 3D landmarks to the meshes. The scanned objects preserve the original scale, with 1 unit corresponding to 1 millimeter. According to 3dMD [2], this type of system has a linear accuracy range of 0.2mm or better on a frame-by-frame basis.

A set of 1,417 heads were scanned. The raw scanned meshes can contain sampling and reconstruction artifacts. To drive direct comparisons and maintain standardization, they are registered to a custom artist-generated template mesh of head and neck geometry consisting of 7,306 vertices, which was originally designed for expression details. Each mesh in the dataset and the registration target were annotated with 6 landmarks. R3DS Wrap software[3] was used to register all original scans to the template, aligned using an Orthogonal Procrustes model [1966].

5.3.2 Public Synthetic Heads Generation

We proceed to leverage the proprietary dataset in order to create a public dataset of 100 synthetic faces, which maintains desirable properties such as high quality and demographic distribution. Similarly to Dai et al. [2019], we use a principal component model (PCA) to generate our meshes [Blanz and Vetter, 1999]. Given M pre-registered meshes, each with N vertices and the same connectivity, we can denote each individual scan as $x_i \in \mathbb{R}^{3N}$. We can then compute the mean shape over the set $\overline{x}$, and recenter each subject around this value $\overline{x_i} = x_i - \overline{x}$, with the new set of shapes $\overline{X} \in \mathbb{R}^{M \times 3N}$. Since in this case $M << 3N$, the covariance matrix $\overline{X}^T\overline{X}$ is rank deficient.

We eigen-decompose the covariance matrix, from which we get the basis as well as the corresponding eigenvalues, which we assume are sorted in descending order of magnitude. We select the top K eigenvalues and their corresponding basis vectors $b = \{b_1, ..., b_K\}$, where $b_k \in \mathbb{R}^{3N}$, as principal components for our statistical model. We selected $K = 196$ which explains 99% of the variance in the input dataset.

To generate synthetic heads, we blend L registered meshes. We begin by inversely solving the coefficients for each registered head e_i, where $e_i \in \mathbb{R}^K$ refers to the coefficient vector of x_i over b. We generate a new synthetic set of coefficients $\hat{e} = \sum_{i=1}^{L} w_i e_i$, where $w_i \geq 0$, and $\sum_{i=1}^{L} w_i = 1$. The blended coefficients $\hat{e}$ are used

[2] https://3dmd.com/products/

[3] https://www.russian3dscanner.com/

to generate a new blended mesh $\hat{x} = b\hat{e} + \overline{x}$. To prevent shading artifacts, we generate a denser version of the blended mesh through subdivision by employing two applications of the Loop Subdivision Algorithm [1987], resulting in 116593 vertices per head.

Our PCA model was generated using 3D geometry from the entire private dataset of 1,417 participants (679 females, 738 males, aged between 18 and 65). For each of the 100 output synthetic heads, we blended $L = 3$ unique meshes (a total of 300), with the ratio set to a simple average $w_s = 1/3, \forall s$. These trios were set to belong in the same sex and race demographic category according to the US Census [2021], as shown in Figure 5.2.

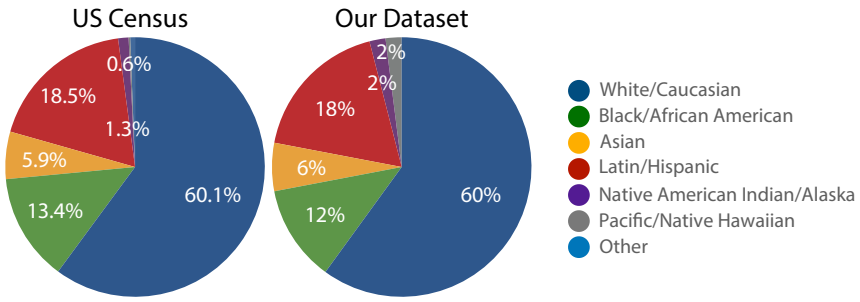


FIGURE 5.2: *Comparison of the demographic distribution showing US census [2021] (left) and our dataset (right). Note that we ensure a minimum of one face is present for each demographic group and gender.*

5.3.3 Distortion Dataset

To study the perception of artifacts on geometry, we needed to generate a set of distorted meshes. We use the synthetic dataset composed of 100 heads described in Section 5.3.2 as our base. We augment this data by modifying it with three selected effects: *noise*, *simplification*, and *smoothing*, as these artifacts are ubiquitous in applications [Botsch et al., 2007]. The parameters of each distortion were carefully selected based on perceptual and quantitative data. A sample set of distorted meshes from our dataset can be seen in Figure 5.3.

Noise

The first artifact we examine is *noise*, which is a common distortion that can stem from many sources in application. Noise can be parametrized by magnitude (how large the deviations are) and frequency (the size of the distortions relative to the mesh). As we want our dataset to be application-agnostic, we chose to add synthetic noise rather than noise stemming from a particular source. We use Perlin noise [2002] due to its simplicity, and because both frequency and magnitude can be easily controlled. We used a three-dimensional noise implementation provided in the SideFX Houdini[4] software suite.

Despite the flexibility of the Perlin noise framework, we need to select a discrete number of frequencies and magnitudes to evaluate in our study. To do this, we begin by searching for an upper bound on the noise frequency. In practice, high-frequency noise often stems from compression algorithms, and is especially notable in methods using direct coordinate quantization of vertices [Deering, 1995]. To estimate the frequency profile of this type of noise, we run an optimization that approximates a

[4] `https://www.sidefx.com/`

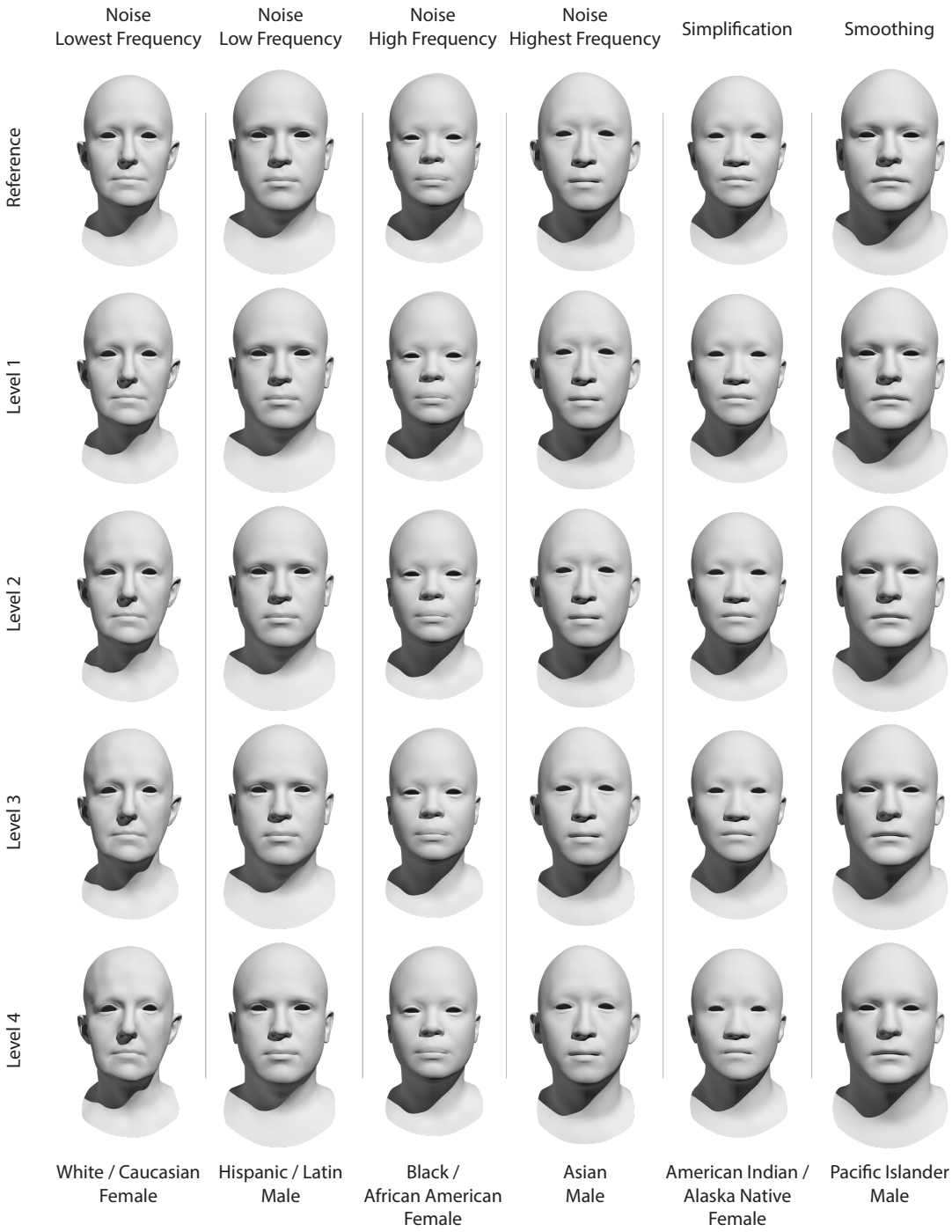


FIGURE 5.3: *Sample of distortions and demographic groups from our dataset. Each column represents a different distortion type and a different demographic group. Rows show sequentially increasing magnitude levels of distortion, starting from an undistorted reference (top), and ending with the highest magnitude of distortion (bottom). Please zoom in to fully appreciate the distortions. Note that in our study an individual face filled about two-thirds of the subjects' displays vertically.*

compressed sphere with a sphere distorted using Perlin noise at varying frequencies. Compression is done using the open-source Draco library[5]. As meshes typically consist of locally varying vertex densities, we run several comparisons on spheres with different, but uniform vertex densities. Simple cost functions like RMSE do not converge, as features caused by compression and noise do not coincide spatially. Instead, we crafted a custom cost function based on the variance of Gaussian curvature, composed of two terms. The first of them, $term_{dist}$, constrains the amplitude (distance term) and the second, $term_{var}$ constrains the frequency of the noise (variance term):

$$cost = a * term_{dist} + b * term_{var}, \tag{5.1}$$

where:

$$term_{dist} = abs(mean(d(S_O, S_C)) - mean(d(S_O, S_N))). \tag{5.2}$$

$$term_{var} = abs(variance(g(S_C)) - variance(g(S_N)), \tag{5.3}$$

Parameters a and b allow us to set the relative importance of the two terms, $g()$ is the Gaussian curvature of the surface, and $d()$ is the Euclidean distance between corresponding vertices of the surfaces. S_O, S_C, S_N are vertices of the original (undistorted), compressed (ground truth), and noisy (optimized) spheres respectively.

A result of this optimization is shown in Figure 5.4. We found that Perlin noise ranging in frequency from 0.5-2 cycles/mm corresponded to plausible fits across all examined densities. We proceed to set our frequency upper bound to 2 cycles/mm. A lower bound of 0.01 cycles/mm was selected manually based on a comparison with the spectral compression algorithm presented by Sorkine and colleagues [2003]. Their work specifically aims at preferentially distorting lower frequencies of a mesh, which are considered by the authors to be less visible. We proceed to interpolate between these two bounds in log space to obtain the four sample points used in our experiment (0.01, 0.06, 0.34, and 2 cycles/mm). A sample set of each frequency and amplitude of noise can be seen in Figure 5.3.

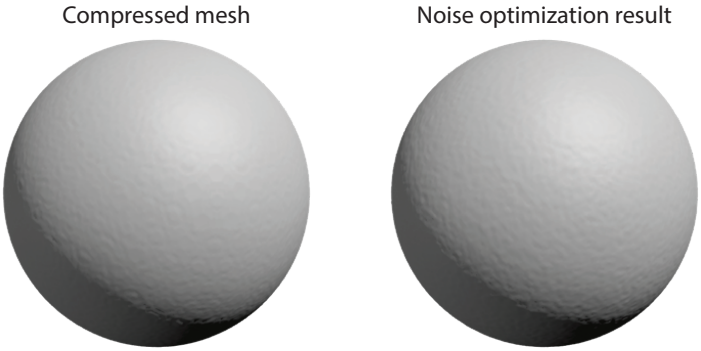


FIGURE 5.4: *Comparison of a compressed sphere (left) and the optimized noisy sphere match (right), which mimics the characteristics of the compression distortion using Perlin noise.*

Next, we need to set the noise amplitudes. Our aim is to select amplitudes that are neither obvious nor invisible, as either would yield only trivial data in the study. To begin, we manually selected a noise amplitude that is clearly visible. We then generate 101 intermediary values by linearly interpolating between this value and the noise-free baseline. Next, we ran a small-scale pilot user study (N=2) for each noise frequency. The experimental setup was identical to the one used in our main study (see Section 5.4.2). We used a Minimum Expected Entropy Staircase experiment

[5] `https://google.github.io/draco/`

procedure [Saunders and Backus, 2006], as implemented in Psychtoolbox[6], to find the amplitude that corresponds to the perceptual threshold (i.e., 1 just-objectionable-difference, JOD, away from the reference, as explained in Section 5.4.4). In addition, the staircase procedure generates a psychometric curve (shown in Figure 5.5). We observe that the curve saturates at approximately 3 JODs. We sample four linearly spaced points between this saturation point and the reference, which guarantees a perceptually significant range of values will be presented to participants. In total, this brings the number of noise trials to 16 (as each frequency is studied at four unique amplitude levels).

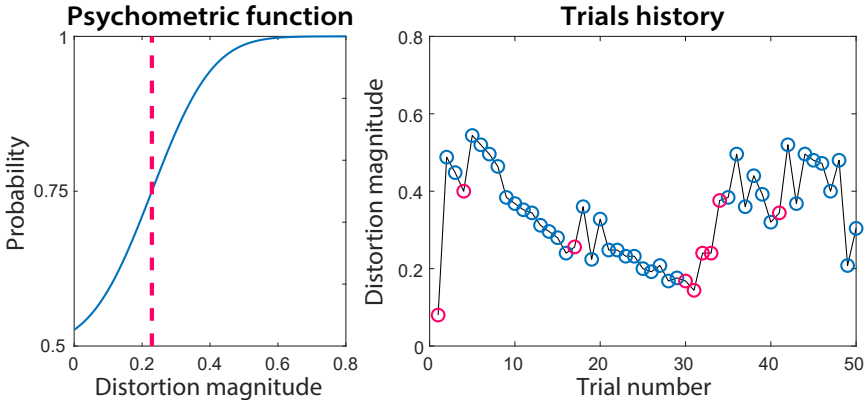


FIGURE 5.5: *A pilot experiment was used to set the distortion magnitudes for our study. We show the user's responses converging towards the threshold in a staircase-type procedure on the right. Blue circles denote a correct answer, while pink ones show mistakes. On the left, a psychometric curve (continuous line) is fitted to the data, with a dashed line marking the threshold value of 1 JOD.*

.

Simplification

The next effect we explored was *simplification*, which is relevant in the context of key applications like decimation, or level-of-detail rendering algorithms.

To select magnitudes of distortion, we followed the simple rule-of-thumb used in many LOD algorithms—reducing triangle count by half at every step [Luebke et al., 2003]. Next, we performed a staircase experiment identical to the one described in Section 5.3.3. We found that a single reduction level (i.e. 50%) was well-below the perceptual threshold. As a consequence, we selected the first level of simplification as 25%, and followed the halving rule to set four magnitudes of simplification (25%, 12.5%, 6.25%, and 3.125%).

Smoothing

The third and final artifact we explored was *smoothing*. In applications, it can stem from simplification, errors during mesh reconstruction, mesh decimation, volume-based remeshing algorithms, etc. To simulate different magnitudes of smoothing, we employed a Laplacian Smoothing algorithm [Witkin, 1987] with fixed boundaries, as implemented in SideFX Houdini. The default value of $\lambda = 0.5$ was used.

In order to select the magnitude of smoothing, we once again employed a staircase pilot experiment (identical to Section 5.3.3). The expert-led experiment showed even

[6] `http://psychtoolbox.org/`

TABLE 5.1: *Parameters of the Experimental Conditions*

Distortion	Magnitude measure	Magnitude			
		Level1	Level2	Level3	Level4
Noise - lowest freq.	amplitude	5.00	8.33	11.66	15.00
Noise - low freq.	amplitude	0.21	0.35	0.49	0.63
Noise - high freq.	amplitude	0.0675	0.1125	0.1575	0.2025
Noise - highest freq.	amplitude	0.0740	0.1233	0.1727	0.222
Simplification	% of remaining triangles	25.0	12.5	6.25	3.125
Smoothing	# iterations	2	6	12	20

a single application of Laplacian smoothing as above the threshold of noticeability. However, unlike other distortions, further testing with naïve users found they were significantly less sensitive to the artifact, which led us to set the first level of magnitude at 2 repetitions. As more aggressive smoothing is needed to produce changes on a smoother surface, we set the remaining values based on extensive pilot experiments. The final values used in our main study are 2, 6, 12, 20 repetitions of Laplacian smoothing.

Overview

We begin with 100 reference meshes of faces, which are distorted using 6 different types of artifacts (4 types of noise, smoothing and simplification). Each artifact has 4 levels of magnitude, resulting in 24 distorted meshes per reference, or 2400 distorted meshes in total. A breakdown of the exact distortions used can be seen in Table 5.1.

5.4 User Study

In order to gather perceptual data, we ran a large-scale user study, detailed in this section. This is the largest psychophysical collection of geometric distortions ever obtained, and is composed of 84,000 subjective comparisons.

5.4.1 Participants

In observance of restrictions for in-person work during the COVID pandemic, this study was conducted remotely. 50 male and 50 female paid participants were recruited (aged between 18 and 55 years). Our study received ethical approval through a third party institutional review board, and each of the 100 paid participants signed an informed consent form. Experimental output was anonymized.

Each experiment session consisted of 280 randomly selected trials over a batch of 10 unique heads. Our dataset of 100 synthetic head meshes was randomly split into 10 batches of 10 meshes each, with 10 participants assigned to each batch. In pilots, participants needed approximately 10 seconds per trial, resulting in an estimated 47 minutes required to finish the main portion of each session, which was deemed adequate to avoid fatigue. Each participant completed 3 experimental sessions each on different days.

5.4.2 Stimulus Rendering

Our experiment aims to measure the perceived strength of distortions, but additional factors like illumination and texture can be expected to influence the result [O'Shea et al., 2008; Serrano et al., 2021]). As there are too many possible configurations of texture and illumination, we choose to perform our experiment using representative conditions that are expected to maximize the subjects' overall sensitivity, following guidelines set in the literature [Rogowitz and Rushmeier, 2001; Lavoué et al., 2016]. We use a distant top-right light source, mimicking natural environmental conditions [Sun and Perona, 1998; O'Shea et al., 2008]. Because this can leave the lower portion of the head unlit, we add a lower intensity light from the bottom-left direction, avoiding unnaturally strong shadows. We do not use any texture, with the heads rendered with a single uniform neutral gray color. This is expected to maximize sensitivity to distortions, as overlapping signals from texture can cause masking [Ferwerda et al., 1997b].

Our experimental software was implemented using the Unity game engine[7] and rendered in real-time on the participants' machine using pre-processed meshes. Participants were allowed to freely rotate the models around the vertical axis, and were instructed to make a holistic decision based on the distortions on the entire mesh. All three meshes rotated in synchrony to enable direct comparisons. When no participant interaction was detected for a few seconds, the meshes began to slowly rotate, as pilot studies suggested this helps users detect distortions.

In addition to the artifacts described in Section 5.3.3, we decided to study two additional ancillary variables. The first is *distance*. Although distance is known to play a large role in the perception of visual artifacts [Daly, 1992a; Ye et al., 2019], it is not modeled by most existing geometry metrics. We study two distance conditions, "close" and "far" (subtending approx. 18 and 9 visual degrees respectively, assuming a standard distance of 2.5 picture heights away from the display as instructed in training). The second variable relates to human sensitivity to faces, in contrast to other objects. Prior art points to humans being especially sensitive to deformations on faces [Zhao et al., 2003], and evidence suggests a higher order holistic process takes place. In order to quantify the sensitivity differential between faces and general objects, we need to compare identical objects, but have one object be perceived as a face, while the other is not. This led us to the influential Thatcher Illusion [Thompson, 1980], which demonstrated that the recognition of significant distortions of a face, such as rotating the eyes and mouth 180°, can go undetected when the face is presented upside-down. This effect has previously been successfully exploited in the computer graphics community to increase a subjects' sensitivity to colormap calibration [Kindlmann et al., 2002], where users presented with upside-down faces were found to be significantly less sensitive to distortion. We follow this template, and present our experimental subjects with two conditions: "normal" and "flipped", where the latter case shows the face upside-down. In this way, for each condition described in Section 5.3.3, we study four cases (close/up, close/flipped, far/up, and far/flipped).

5.4.3 Experimental Procedure

We chose to employ a two-alternative forced choice (2AFC) procedure, as this is considered the most accurate way to measure threshold distortion visibility [Perez-Ortiz et al., 2019]. A sample trial of our study is shown in Figure 5.6. The participant

[7] `https://unity.com/`

TABLE 5.2: *Comparison Matrix*

	Reference	Level 1	Level 2	Level 3	Level 4
Reference		✓	✓		
Level 1	✓		✓	✓	
Level 2	✓	✓		✓	✓
Level 3		✓	✓		✓
Level 4			✓	✓	

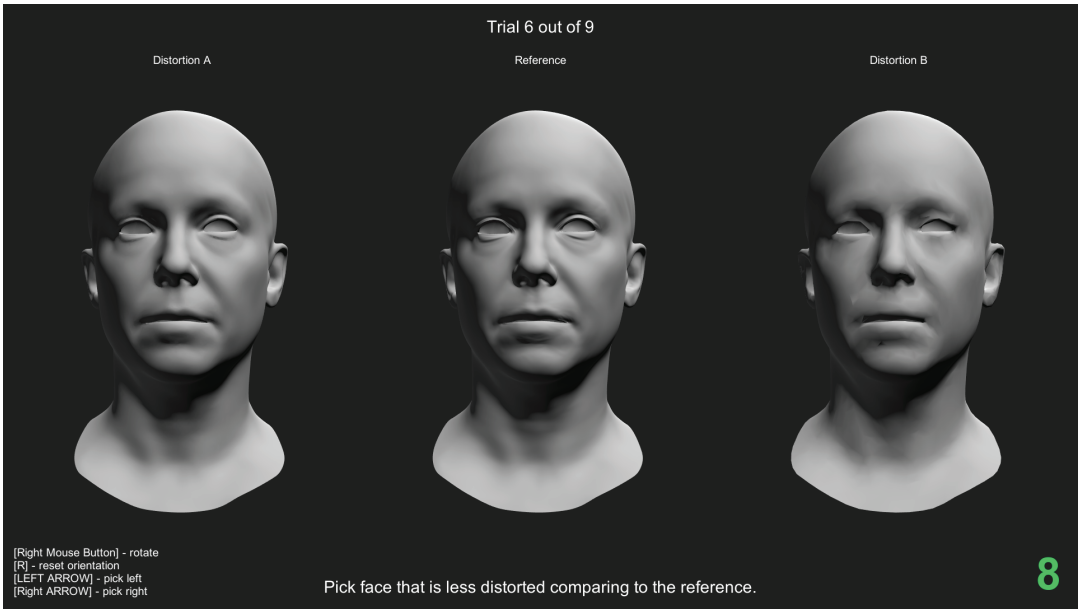


FIGURE 5.6: *A screenshot of the software used in our subjective study. The observer is presented with 3 heads at a time: a reference placed in the center, and two distorted heads randomly assigned to either side. The task is to pick the head that is less distorted in relation to the reference by pressing the left or right arrow keys.*

is presented with an unmodified reference face, selected from the synthetic dataset described in Section 5.3.2, and two variations subject to a the same artifact at different magnitudes (as described in Section 5.3.3). The subject is prompted to select the head that is less distorted compared to the reference by pressing on the left or right arrow keys.

To avoid obvious comparisons which do not add information to the set, only distortion magnitudes of up to 2 steps away from each other were compared [Perez-Ortiz et al., 2019], as shown in Table 5.2.

Before beginning each experimental session the subjects were trained. A representative subset of comparisons were shown containing all distortion types with exaggerated magnitudes, simplifying the task while avoiding over-training. During training, participants were given feedback on whether their answer was accurate to help accustom them to the experimental procedure. A separate set of meshes that was not repeated in the main study was used for training, and training results were discarded from further analysis.

Participants performed the experiment at home, and were instructed to sit comfortably at an approximate distance of 2 to 3 picture heights from their display. Breaks were allowed during the study, and it was possible to pause or exit the experiment and pick progress up at a later point if necessary. In order to avoid users getting stuck on difficult comparisons, we implemented a 30 second timer on the bottom right, and

instructed participants to try not to exceed this time limit per trial, although this rule was not enforced. A minimum trial time of 2 seconds was enforced in order to avoid accidental responses. Once a session was completed, the anonymized results were automatically uploaded to our servers.

5.4.4 Data Analysis

To allow for uniform comparisons across distortion types and magnitudes, we convert users' 2AFC responses using perceptual scaling (see Section 2.3.1). To do this, we employ the software developed by Perez-Ortiz et al. [2019] for each of the studied distortions. This results in a per-artifact JOD scale, which can be directly used for perceptually meaningful comparisons and predictions. This method also allows for automatic statistical outlier removal, which flagged 7 participants whose results were discarded from further analysis.

5.5 Experiment Results

The results of our experiment are shown in Figure 5.7. A separate plot is presented for each of the six distortions studied (4 frequencies of noise, simplification and smoothing). On the horizontal axis, we show progressively larger magnitudes of distortion. On the vertical axis, we show a JOD-scale distance from the reference. The four lines represent the results of each combination of ancillary conditions (far/near, normal/flipped). Vertical bars show 95% confidence intervals, calculated using bootstrapping, as described by Perez-Ortiz et al. [2019]. Each plot shows the mean results across all participants. Below, we discuss the results for each distortion.

Noise The results in the first four plots (two first rows) correspond to the distortion caused by *noise*. First, we can focus on the top left plot, showing the lowest frequency. Note that, as expected, the JOD distance from the reference increases gradually as higher magnitudes of distortion are examined. In an effect only seen at the lowest frequency, users were more sensitive to the *far* conditions. We theorize this is due to the human visual system being less sensitive to stimuli at both very low and very high frequencies [Campbell and Robson, 1968]. As the pattern moves away from the viewer, it subtends a smaller area resulting in an increase in perceived frequency as experienced by the observer, and consequently an increase in sensitivity for this distortion. This effect was previously exploited by Oliva and colleagues in their work on "Hybrid Images" [2006]. As the noise frequency is increased in the subsequent plots (top right, middle left, and middle right), we see the near and far conditions reverse, making the far condition less visible than near, with the separation becoming more apparent at higher frequencies. This follows the same insight, with sensitivity falling off for high frequencies which appear even higher to the observer due to increased distance, resulting in lowered sensitivity.

Simplification The bottom left plot shows the result for *simplification*. As expected, a significant difference between the close/far conditions is observed. This data follows the perceptual insight leveraged for popular level-of-detail rendering methods, which simplify meshes that lie further from the observer.

Smoothing The result is shown in the bottom right plot. *Smoothing* was much less visible to our observers than the other distortions, with a maximum mean JOD

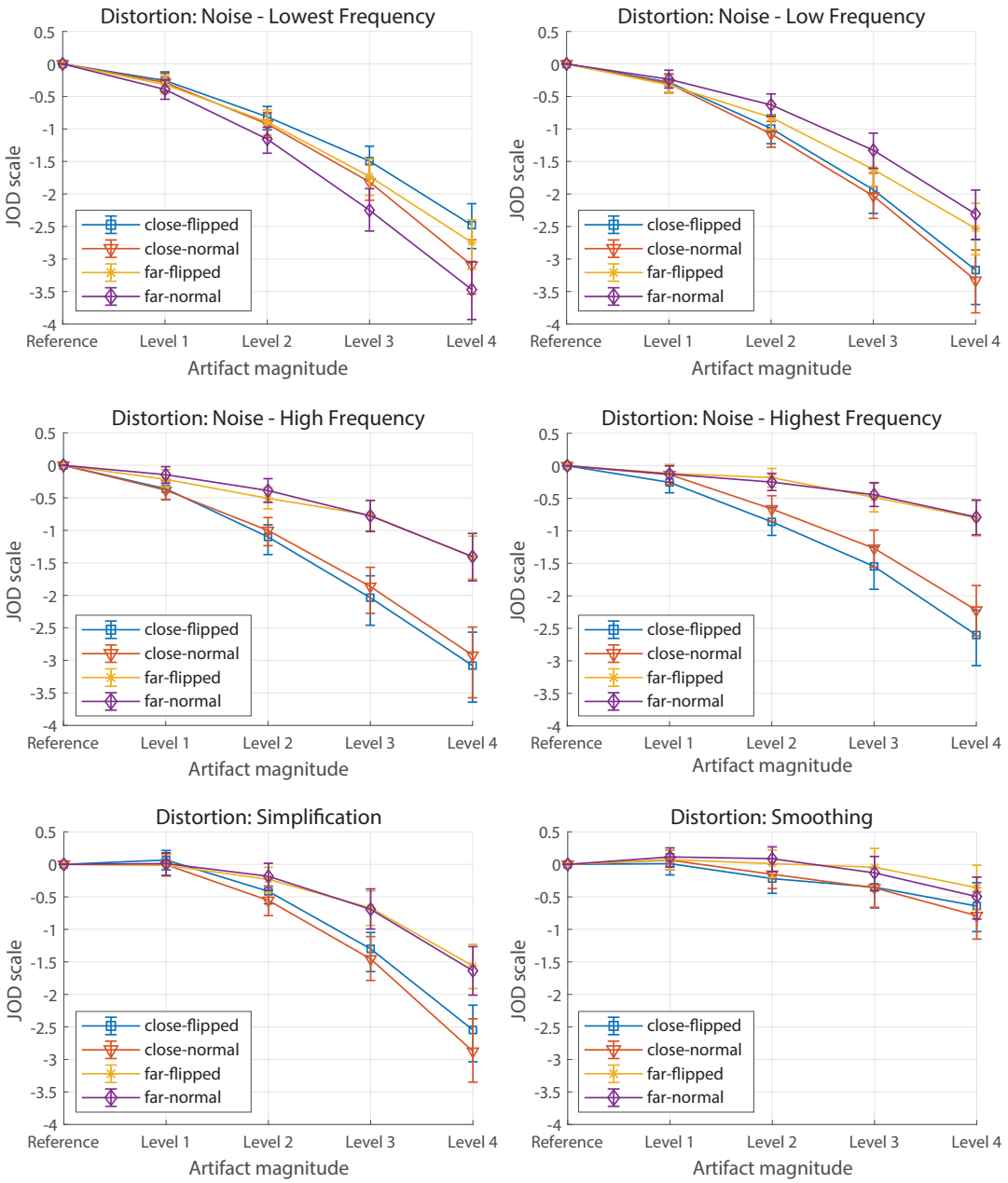


FIGURE 5.7: *This figure shows the results of our subjective study. Each plot represents one of the six distortions we measured. Every line corresponds to a separate condition. Note the strong impact of distance for the three highest noise frequencies and simplification. Please see our analysis in Section 5.5.*

distance of only -0.79. This is likely due to the localized nature of the distortion, which is only visible in areas of the mesh that contain high frequencies. In addition, our dataset of relatively smooth head meshes may also have led to more difficult comparisons.

Overview We generated a perceptually meaningful scale of distortion for each of six studied artifacts. The effect of *distance* was strongly present in all conditions, either enhancing or obscuring the visibility of distortions following expected relations from contrast sensitivity models [Nader et al., 2015]. Conversely, the normal/flipped conditions did not impact responses. We hypothesize this is due to the unlimited length of presentation, which may have allowed participants to mentally cancel the effect of presenting a face upside-down [Lewis, 2001]. We confirm these findings through a statistical analysis of the results using the effect size d [David, 1963]. As a consequence, in the following sections, results are averaged over the normal/flipped conditions.

5.6 Distortion Metric Analysis

We begin by analyzing the performance of popular metrics on 3D geometry: RMS, Metro [Cignoni et al., 1998], DAME [Váša and Rus, 2012], MSDM [Lavoué et al., 2006], MSDM2 [Lavoué, 2011] and FMPD [Wang et al., 2012]. More information about the metrics can be found in Section 2.2.2.

As artifacts differ significantly in their subjective characteristics, it is unlikely that any metric is able to successfully explain the entire dataset simultaneously. We confirm this by calculating Pearson's Linear Correlation Coefficient (PLCC) and Spearman's Rank Correlation Coefficient (SROCC) compared to the results of our subjective study, which can be found in Table 5.3. All demonstrated coefficients demonstrate low correlation with collected subjective data.

In order to obtain more targeted performance indicators, we proceed to study the correlation of each metric to each of the studied distortions separately. DAME and MSDM are excluded from the *simplification* trials, as these metrics require vertex-to-vertex correspondence. We begin the analysis using only the results of the "close" condition, shown in Table 5.4. FMPD shows the best performance for *noise*, with top PLCC results in every case. Metro, MSDM, and RMS obtain the best SROCC for low, high, and highest frequency noise, respectively. Surprisingly, the simplest metric RMS performs best for *simplification* and *smoothing*, with the latter artifact having the worst overall scores across all metrics.

Table 5.5 shows the same analysis performed for the "close" and "far" conditions together. As none of the metrics model distance, which is a significant factor to the subjective judgments, performance drops happen for most comparisons highlighting the need for metrics that incorporate this perceptual factor. DAME now has the best PLCC performance for lowest-frequency noise, while FMPD maintains the best performance for low, high, and highest-frequency noise. RMS remains the best PLCC performing metric for *simplification* and *smoothing*.

5.7 Applications

In this section, we show some practical applications that demonstrate the utility of our perceptual dataset.

TABLE 5.3: *Correlation Coefficients Computed For Merged Distortions*

	All conditions				Close heads			
	All distortions		Without simplification		All distortions		Without simplification	
	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC
METRO	-0.30109	-0.47088	-0.30968	-0.49465	-0.22184	-0.45389	-0.22283	-0.47155
DAME	NA	NA	-0.41877	-0.54464	NA	NA	-0.56515	-0.67102
FMPD	-0.29465	-0.41442	-0.23443	-0.42364	-0.41297	-0.53994	-0.40786	-0.56354
MSDM	NA	NA	-0.584	-0.5784	NA	NA	-0.59759	-0.63093
MSDM2	-0.47868	-0.5129	-0.48726	-0.54103	-0.57896	-0.61081	-0.58775	-0.63955
RMS	-0.35873	-0.51673	-0.3901	-0.50025	-0.25933	-0.5014	-0.28432	-0.46842

TABLE 5.4: *Correlation Coefficients Computed for Each Distortion Type Separately (for Close Heads)*

	Noise								Simplification		Smoothing	
	Lowest freq.		Low freq.		High freq.		Highest freq.					
	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC
METRO	-0.83786	-0.85531	-0.85159	-0.8886	-0.80548	-0.85085	-0.73139	-0.76049	-0.81164	-0.80853	-0.54	-0.50158
DAME	-0.83952	-0.86269	-0.83788	-0.87335	-0.78928	-0.81867	-0.72004	-0.74596	NA	NA	-0.54911	-0.53409
FMPD	-0.85387	-0.87492	-0.87582	-0.87283	-0.83075	-0.8595	-0.73603	-0.75207	-0.82596	-0.7967	-0.43741	-0.4882
MSDM	-0.81545	-0.86174	-0.75198	-0.87583	-0.71318	-0.83951	-0.61431	-0.77115	NA	NA	-0.50298	-0.52468
MSDM2	-0.7756	-0.86293	-0.8146	-0.88512	-0.76524	-0.84845	-0.64805	-0.75457	-0.66364	-0.80923	-0.55012	-0.52745
RMS	-0.8377	-0.85725	-0.85082	-0.88765	-0.80674	-0.87288	-0.72701	-0.73285	-0.82752	-0.79958	-0.55147	-0.54268

TABLE 5.5: *Correlation Coefficients Computed for Each Distortion Type Separately (for Both Distances)*

	Noise								Simplification		Smoothing	
	Lowest freq.		Low freq.		High freq.		Highest freq.					
	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC
METRO	-0.84552	-0.87899	-0.80067	-0.83565	-0.64015	-0.65671	-0.56912	-0.58581	-0.71671	-0.70817	-0.47613	-0.44695
DAME	-0.8503	-0.88571	-0.78221	-0.80956	-0.63214	-0.6478	-0.55242	-0.55489	NA	NA	-0.47313	-0.46808
FMPD	-0.83415	-0.87444	-0.8242	-0.82314	-0.66238	-0.66313	-0.57095	-0.57861	-0.73287	-0.70548	-0.3836	-0.45222
MSDM	-0.81724	-0.86826	-0.70719	-0.818	-0.56928	-0.66113	-0.47542	-0.58426	NA	NA	-0.43856	-0.4754
MSDM2	-0.77799	-0.86893	-0.76497	-0.8251	-0.61003	-0.65798	-0.50469	-0.59126	-0.57988	-0.71234	-0.47943	-0.46324
RMS	-0.84133	-0.86611	-0.79922	-0.83484	-0.64533	-0.67854	-0.56143	-0.54794	-0.73604	-0.70532	-0.49102	-0.4832

5.7.1 Retargeting Metrics

We build on the analysis of geometry metrics performed in Section 5.6 by retargeting these methods to fit our perceptual data. These metrics produce results in different, non-perceptual units. This makes it difficult to use their output to perform commonly encountered engineering tasks, such as setting meaningful thresholds or stopping criteria, as the output values have no perceptual reference. Typically, this type of parameter can be set through experiment, but relating the results of an experiment to the metric output can be challenging, and thresholds often do not generalize beyond the original results. This highlights the importance of perceptually scaled data, such as our dataset, to achieve practical tasks. We give a concrete example of this scenario for mesh compression in Section 5.7.2.

We also observe that distance, despite being a major perceptual factor to the visibility of artifacts, is not modeled by any of the examined metrics, which produce identical responses regardless of the users' position in relation to the object. We explore an extension of metrics to distance using our dataset for a LOD application in Section 5.7.3.

New perceptually-aware metrics for geometry distortion could be developed to address the points above, but generating such a metric is beyond the scope of this work. We simplify the problem by instead retargeting the responses of the metrics we examined in Section 5.6, allowing us to inject some perceptual insights. After experimenting with different functional forms (linear, polynomial, and exponential), we observed that an empirical remapping of the metrics' output to the perceptually-motivated JOD units by applying scaling in the form of the exponential function $Y = a \cdot X^b$ produces the best results of all examined models. This produced good fits of metrics' responses to our subjective data for a single distance, where X is a raw

metric response, Y is metric retargeted to the JOD scale, and a and b are optimized fitting parameters.

We also explicitly incorporate distance as a factor to the metric responses. We do this by modulating the results using the ratio of visual angles subtended by the object. We set the size of the stimuli shown in the *close* condition of our study as the reference α, and the current size of the object as β. We then modify our retargeting formula to $Y = \frac{\beta}{\alpha} \cdot a \cdot X^b$. As is often practical in application, we can reformulate this to use distance relative to the observer. We define m_e as the maximum extent of the object (half the maximum size of the bounding box), d_1 as the distance corresponding to the default visual angle α, and d_2 as the distance for which we are querying (schematized in Figure 5.8). This results in:

$$Y = \frac{\arctan(m_e/d_2)}{\arctan(m_e/d_1)} \cdot a \cdot X^b \tag{5.4}$$

The values of the parameters a and b are fitted to the dataset, minimizing error between the remapped metric response and JOD values found in our experiment for each stimulus. In addition, we add a boundary condition so that large distances ($\alpha < 0.11°$) have a score of 0 JOD as distortions are not visible at that scale. Based on the conclusions of Section 5.6, we perform this fitting separately for each metric and distortion type.

Through Equation 5.4, we can use legacy metrics to make predictions that account for distance, and approximate metric output to the JOD units used in our study. In the following sections, we apply a retargeted version of FMPD to compression and LOD rendering.

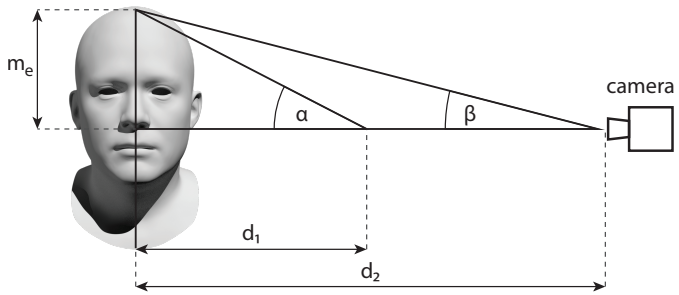


FIGURE 5.8: *The trigonometric relationship between visual angle, size, and distance from the observer used in our model.*

5.7.2 Perceptually Bound Compression

When employing mesh bitrate compression, a cost vs. quality trade-off is present—the goal is to compress at the lowest rate that presents acceptable subjective quality. To test the performance of the metric retargeting performed in Section 5.7.1, we ran a validation study.

Stimuli Although our main study consisted only of human faces, we wanted to explore the hypothesis that our learnings can generalize by selecting a varied set of test cases. These consisted of shapes used by Lavoué et al. [2006] (venus, armadillo, rocker), as well as one face from our dataset. The former present challenges, as these shapes are very different from those we studied. In particular, noisy references, such as the armadillo mesh, can result in reduced visibility of artifacts due to masking,

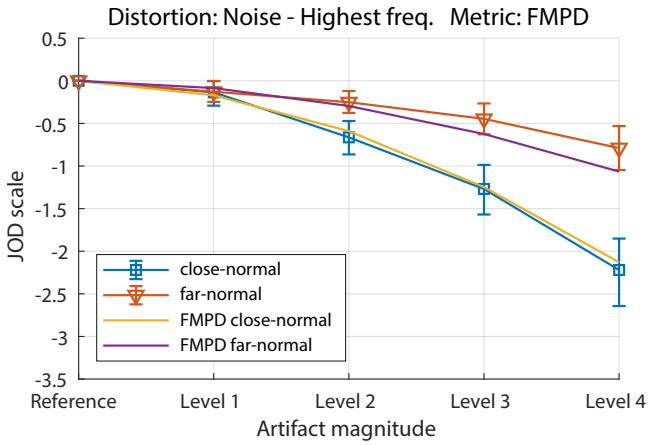


FIGURE 5.9: *We retarget FMPD as described in Section 5.7.1 to assess distortions caused by compression algorithms. Orange and blue lines represent the results of our study for the close and far conditions, while yellow and purple lines correspond to the remapped output of FMPD.*

which was not modeled in our work. For each object, we generate a progression of quantized meshes using the Draco library. Note that some low-compression-factor meshes were excluded during piloting, as they were deemed visually identical to the reference.

Procedure We used a setup identical to that described in our main experiment (Section 5.4.3), with 8 additional participants. Participants' responses were converted to JOD units using the method of Perez-Ortiz et al. [2019]. We compared these values to the retargeted version of the FMPD metric, shown in Figure 5.9.

Results The results of our validation can be seen in Figure 5.10. Note that, unlike traditional metrics, our retargeted metric's output can be directly compared to the results of the subjective study. The metric matches the perceptual outcome well—with identical selections for 3 out of 4 meshes at a threshold level of 1 JOD.

5.7.3 Perceptually Driven Automatic LOD

Level-of-detail rendering is a popular real-time graphics technique, swapping out meshes for lower complexity alternatives when objects are distant in order to improve performance. Despite its ubiquity, most practical implementations rely on distance-swapping criteria set manually by artists. Traditional geometry metrics such as those examined in Section 5.6 are not distance-aware, preventing their use for this application. Conversely, a metric trained on our distance-aware data could estimate visibility based on distance, generating an automated quality-cost trade-off prediction. To exemplify this, we use a retargeted version of FMPD (as described in Section 5.7.1).

For a given mesh L_0, we begin by pre-computing a set of simplified versions $L_{1..N}$. We then solve an inverse problem, finding the distance d_i at which the difference between the reference and each of the simplified meshes lies beyond a given perceptual threshold value δ at each step, i.e. $d_i = \text{argmin}_d(|L_0 - L_i|_\text{d} \leq i\delta)$.

We compare the prediction of our retargeted metric on a head from our dataset to a manual setting chosen by an artist in Unity, who was instructed to pick the earliest point that avoided visible artifacts. We show the results of our method in Figure 5.11. Note that our model is built using side-by-side comparisons, while the visibility of popping artifacts in an LOD setting is more similar to a flicker comparison, which is

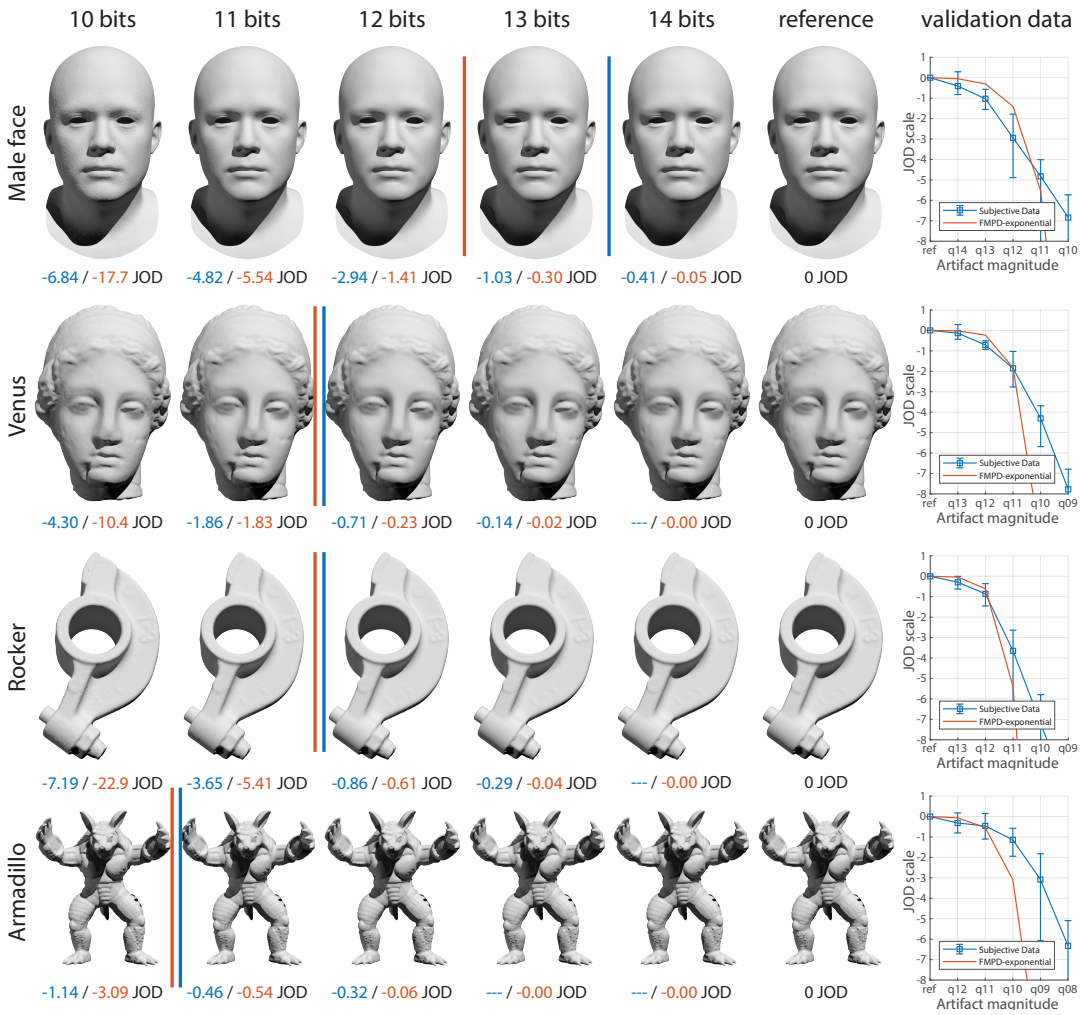


FIGURE 5.10: *A look at our compression application. Each column represents a mesh, encoded with a static number of bits per coordinate. The rightmost column shows the results of our validation experiment (Section 5.7.2) contrasted to the output of the retargeted FMPD metric, also shown under each mesh. Blue and orange separators denote the perceptual threshold, set to 1 JOD (corresponding to a 75% probability of preference), according to the study (blue) and metric (orange). Note that in contrast to our work, setting a meaningful threshold in arbitrary metric units is a non-trivial task. Please zoom in to observe the distortions.*

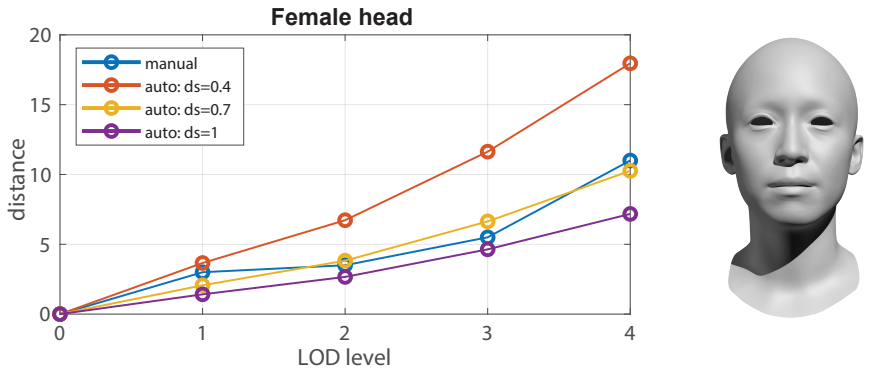


FIGURE 5.11: *Our retargeted metric can be used to automatically predict the distance at which an LOD swap should occur. Here, we compare our estimate to a selection made by an artist instructed to avoid visible popping artifacts (left) and present the mesh used for the evaluation (right).*

expected to be significantly more sensitive [Simons and Rensink, 2005]. As a result, using the same threshold of 1 JOD as Section 5.7.2 leads to swaps at lower distances than those selected by the artist. However, a lower threshold of 0.7 JOD produces a similar swapping profile. A perceptual data-driven model like the one presented here can be used by artists to automatically set an application-wide subjective threshold, reducing manual labor in selecting LOD settings.

5.8 Conclusions

In this work, we explore the perceptual impact of distortions on 3D geometry. We created a novel synthetic dataset, consisting of 100 head scans which mimic the demographic distribution of the US. We identified geometric distortions of interest to relevant applications, and ran carefully crafted pilot studies and numerical simulations to set the appropriate parameters for our perceptual study. We gathered a large-scale dataset of subjective responses for geometric distortion, including relevant perceptual parameters, such as distance. We used the results of our study to analyze the performance of geometric metrics. We also explored applications of our dataset to use cases like metric retargeting, automating stopping criteria for mesh compression, and LOD rendering.

Limitations and Future Work Although we collected a large dataset of perceptual data on geometry, we were only able to sample a limited number of parameters and modalities for each distortion. In particular, it would be interesting to further explore the impact of distance on perception by sampling more in this dimension. Our dataset was composed of meshes representing human faces, which we measured in a normal and flipped orientation with the intent of comparing the perceptual impact of human sensitivity to faces. Although we found no difference in sensitivity between the two, it would be interesting to add a time-limited component to these trials to make it more difficult for users to mentally invert the rotation. Early piloting of our study with meshes representing arbitrary objects indicates that increased sensitivity for faces is present, and quantifying this numerically could generate valuable insights. Further, although masking is known to be a major effect for the visibility of perceptual artifacts, we did not explore it in this study. Notably, our study was done on meshes without texture, which is a likely source of masking for the distortions studied in this work. Future efforts should be made to collect psychophysical data on geometric distortions while actively controlling for masking due to the properties of the reference mesh and texture.

Chapter 6

Perceptual Model for Adaptive Local Shading and Refresh Rate

6.1 Introduction

Shading is one of the most computationally expensive parts of the rendering pipeline, yet the demand for shading computations is growing significantly with the increase of both the resolution and refresh rate of displays. Even the most powerful GPUs are unable to meet these demands and are limited by their computational power and bandwidth. Furthermore, with the growing popularity of mobile gaming, which needs to operate under a limited power budget, and also GPU sharing cloud gaming [Yadav and Annappa, 2017], rendering often needs to operate at a fraction of the maximum GPU capacity. To address this challenge, all popular GPU manufactures have introduced a more flexible shading mechanism called VRS in their next-gen chipsets [Nvidia, 2018; AMD, 2021; Qualcomm, 2021]. VRS enables the control of the resolution of shading within each 16×16 image tile, while retaining visibility computation at the native resolution. VRS has been used to exploit the limits of the HVS by intelligently distributing the shading budget based on foveation [Tursun et al., 2019], scene content and motion [Yang et al., 2019], or depth-of-field [Intel, 2019]. All these works propose a dynamic quality control mechanism that allocates the rendering budget to those aspects of an image or animation that have the highest impact on the overall quality.

In this chapter, we propose to control both the VRS state map and the refresh rate, based on all major factors affecting image quality: texture content, on-screen velocities, luminance, effective resolution, and display persistence. We build on the work of Denes et al. [2020], and extend the visual quality model to account for the important effect of texture content and VRS resolution. In contrast to that work, we offer local, rather than global, control of the resolution via VRS without the need for eye tracking.

The key component of our Adaptive Local Shading and Refresh Rate (ALSaRR) method is a new Content-adaptive Metric of Judder, Aliasing and Blur (CaMoJAB), which we discuss in Section 6.3. The metric is based on psychophysical models of contrast sensitivity with only a few parameters fitted to the data. We calibrate and validate our metric on various existing datasets as well as our new dataset collected by conducting a subjective quality experiment. The experiment measures the perceived loss of quality due to shading rate reduction under a large range of display refresh rates, resolutions, display persistence, luminance, contrast, and content velocity (Section 6.3.3). Our ALSaRR method uses the new metric to create per-texture quality functions, which are used for an approximate solution of the knapsack problem: maximize perceived quality for a given rendering budget (Section 6.4). Unlike the method of Yang et al. [2019], which controls VRS to avoid any visual loss regardless of

the per-frame rendering cost, our goal is to find the best trade-off of spatio-temporal resolution under a limited rendering budget. The source code of the metric and mentioned dataset are publicly available online[1].

6.2 Background

Existing studies distinguish two main directions of applicability of motion quality metrics: motion-adaptive and fixed-bandwidth rendering. Motion-adaptive rendering seeks to deliver perceptually lossless rendering quality, while fixed-bandwidth rendering aims to provide the best possible user experience within a given rendering budget. This section examines the relevant research related to these two application contexts, providing insights into possible content manipulation methods in order to achieve stated objectives.

6.2.1 Motion Adaptive Rendering

Locally adaptive real-time shading is a relatively new area as the traditional rasterization frameworks did not allow for a sub-image level of control until recently. The major focus of these developments has been on rendering for VR with the goal of either matching the non-uniform pixel distribution of HMDs [Pohl et al., 2015] or exploiting the reduction of visual acuity with eccentricity [Tursun et al., 2019]. The recent introduction of VRS allowed for controlling the shading rates of each 16×16 pixel tiles enabling a plethora of optimizations. Vaidyanathan et al. [2012] analyzed motion and defocus blur in the frequency domain to adaptively control the shading rate and Yang et al. [2019] developed a similar framework better suited for VRS hardware. Drobot [2020] extends this idea to all platforms through their software-based implementation of VRS. Schmid et al. [2019] describe how ray-tracing of reflections can be done in a fashion similar to VRS by varying the number of ray samples for each 8×8 pixel tile. Most of the above techniques are not perceptually motivated and their main focus has been on reducing the rendering cost while yielding visually equivalent output. We argue that the goal of obtaining no visual loss is an impractical target for real-time rendering, as producing high-quality results may require a much larger computational budget than available. In contrast to those works, our aim is to deliver the minimum degradation of quality under a constrained rendering budget.

6.2.2 Fixed-Bandwidth Rendering

Fixed-bandwidth rendering involves the optimal distribution of rendering resources without exceeding a fixed rendering budget given in *pixels/sec*. This can be done by either trading off between resolution and refresh rate [Debattista et al., 2018; Denes et al., 2020] or by fixing the refresh-rate and dynamically changing the resolution [Unreal, 2021; Unity, 2021]. All of these works control the global resolution of a frame and extending them to adapt the resolution locally is non-trivial.

The problem of maximizing quality for a given bandwidth is well studied in the rate-distortion theory but the common solutions involve coding tree units and dynamic programming [Ortega and Ramchandran, 1998] not suitable for real-time rendering. Our proposed optimization is similar to the one proposed by Li et al. [2017] who found the optimal trade-off between delay, power, and rate-distortion in dynamic adaptive streaming applications by formulating it as an integer linear

[1] `https://www.cl.cam.ac.uk/research/rainbow/projects/alsarr`

programming problem and providing a greedy approximation. We derive a similar formal bound on quality more suited for VRS and propose a parallel implementation on GPU that meets the demands of real-time rendering.

6.3 CaMoJAB: Content-adaptive Metric of Judder, Aliasing and Blur

Our goal is to design a motion quality model that, given a display configuration and an image, can predict how the quality will change with display refresh rate, shading resolution, display persistence, and image velocity. We aim to make this model as simple as possible for real-time usage while covering a wide range of display parameters and motion artifacts. The texture content is a major factor influencing quality degradation due to VRS, however, most existing perceptual models ignore the effect of content on motion quality. Therefore, we also aim to make our model content-adaptive.

In the following sections, we develop a motion quality model by studying the evolution of a continuous motion signal and how it goes through various spatial and temporal distortions due to the discrete nature of rendering and display systems. We focus on the distortion that VRS introduces to the mapped texture and we assume that the distortion due to a lower resolution of shading (e.g. specular reflections) is negligible. The following analysis models an image that is stabilized on the retina therefore we do not consider the changes of the signal over time. We define the motion quality q as a negate of weighted sum of spatial distortions (d_s) and temporal distortions (d_t) and show that such a definition correlates well with the experimental results.

$$\begin{aligned} q(t(u,v),v,\boldsymbol{D}) = &- w_\mathrm{s} \cdot d_\mathrm{s}\left(t(u,v),v,\boldsymbol{D}\right) + \\ &- w_\mathrm{t} \cdot d_\mathrm{t}\left(t(u,v),v,\boldsymbol{D}\right), \end{aligned} \tag{6.1}$$

where $t(u,v)$ is the mean luminance of the observed portion of an image in $\mathrm{cd/m^2}$, v is the velocity of the texture movement on the screen in $^{\mathrm{deg}}/_{\mathrm{s}}$, and $\boldsymbol{D}$ is a display configuration parametrized as:

$$\boldsymbol{D} = \{\widehat{\omega}, p, \widehat{\rho}, r_\mathrm{x}, r_\mathrm{y}\}, \tag{6.2}$$

where $\widehat{\omega}$ is the temporal Nyquist frequency of the display in Hz (half of display's refresh rate), p is the persistence (or duty-ratio) of the display ($0 < p \leq 1$), $\widehat{\rho}$ is the spatial Nyquist frequency of the display in cycles-per-degree (cpd) (half of display's spatial resolution in pixels-per-degree (ppd)), and $r_\mathrm{x}, r_\mathrm{y} \in \mathbb{Q}^+ \mid (0 < r_\mathrm{x}, r_\mathrm{y} \leq 1)$ are the shading resolution in x and y direction, respectively. The unit of Q is *Just-noticeable-difference* (JND).

Our model is partially inspired by Watson's analysis of capture and display systems [2013] and Denes et al.'s content-independent motion quality model [2020]. However, we focus on understanding the effect of texture content on motion quality to enhance and guide VRS in real-time graphics. For the rest of the chapter, we will refer to our model as ***CaMoJAB***, a content-adaptive model of judder, aliasing, and blur.

6.3.1 Spatial Distortion Model

In a typical VRS rendering pipeline, a continuous texture goes through several stages before it is displayed (Figure 6.1). The texture is first quantized and stored as a

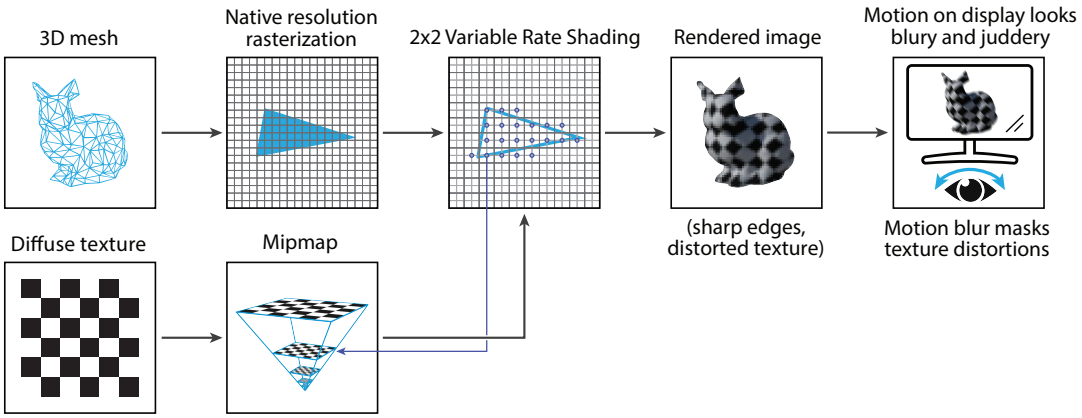


FIGURE 6.1: *Spatial and temporal distortions in a VRS pipeline. Rasterization happens at native resolution but pixel shader is called once per multiple pixels. Larger pixels size leads to selection of coarser mipmap levels yielding distorted texture in the rendered image (artifacts exaggerated for illustration). When the rendered image is displayed at discrete time intervals, the motion appears blurry and juddery. The motion blur may mask the texture distortions caused by VRS.*

mipmap. Next, during the texture lookup, a mipmap level is selected according to the shading resolution specified by VRS. The image is then displayed on the screen for a portion of the frame duration, which depends on the display persistence. If the object is in motion and is tracked by our gaze (smooth pursuit eye motion (SPEM)), it may introduce hold-type or eye motion blur. All the above stages add different kinds of spatial distortions to the signal which results in loss of quality. We will now analyze each of these steps in detail in the frequency domain (Figure 6.2).

Frequency Domain Analysis Analyzing a visual signal in the frequency domain is a useful technique as it facilitates understanding sampling and filtering operations as well as allows for the use of various psychophysical models that are based on spatial frequency theory. Hence, we transform a given continuous 2D texture signal $t(u, v)$ moving with a velocity v in a horizontal direction, into its frequency domain representation $T(\tau_u, \tau_v)$ using *Fourier transform*.

$$T(\tau_u, \tau_v) = \mathcal{F}\{t(u, v)\} \quad [\text{cycles}/\text{texel}], \tag{6.3}$$

where τ_u and τ_v are the spatial frequencies in cycles-per-texel (cpt) corresponding to u and v direction, respectively. Though we assume horizontal velocity in our model, we explain how it can be extended to arbitrary velocity in Section 6.4.

Variable Rate Shading VRS executes the pixel shader once per multiple pixels and replicates the calculated color to all those pixels. The color typically involves a texture look-up operation on the *mipmap* of the underlying albedo/diffuse texture. Mipmapping [Williams, 1983] is the most popular method of antialiasing for textures. A mipmap is a pyramid structure created by low-pass filtering and downsampling to half the resolution of each fine level to create a coarser level. The common filters used for this purpose are box, Gaussian, Lanczos, and Kaiser [Akenine-Möller et al., 2019, Section 6.2.2]. In the following discussion, we assume a box filter as it is one of the most commonly used filters and yet is also known to be one of the worst possible filters because it unnecessarily blurs low frequencies while retaining some high frequencies that cause aliasing [Akenine-Möller et al., 2019]. For every shaded pixel, the area of the pixel is projected onto the texture. The projected area may

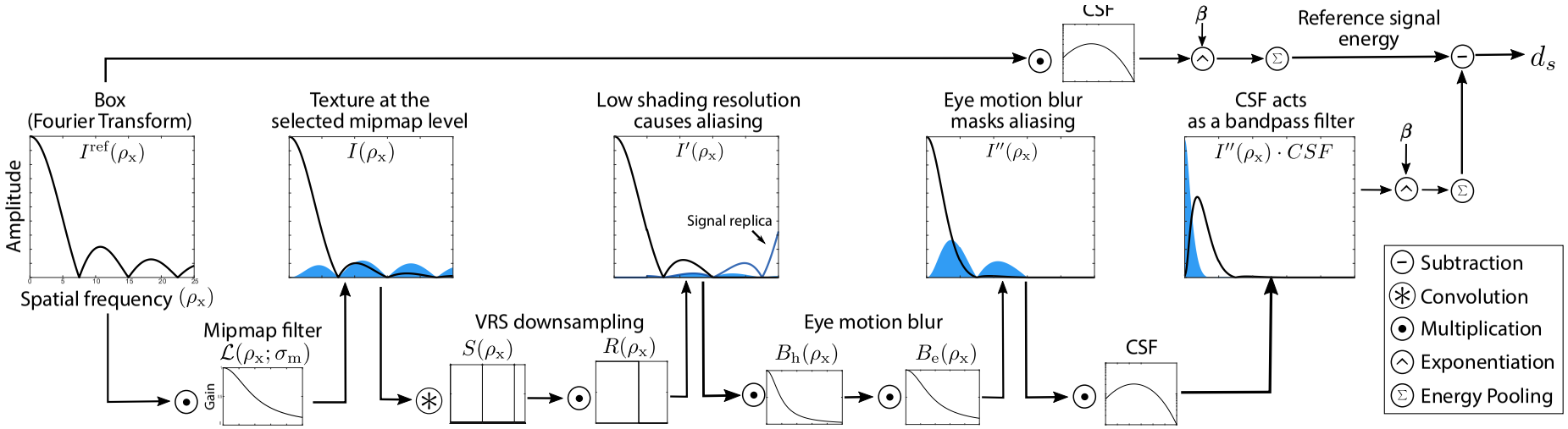


FIGURE 6.2: *Illustration of spatial distortions introduced when an image with a sharp edge (box) is moving on the screen. The black line shows the signal in the frequency domain and the blue shaded region denotes the distortion (difference between the current and previous signal). Picking a low shading resolution leads to a selection of coarser mipmap levels resulting in a loss of high frequencies. Low shading resolution also results in the formation of aliases that distorts the lower frequencies. When the rendered image moves on a screen, our eyes undergo SPEM to follow it. This adds further blurring of the signal and may mask aliasing. The above distortions can be mapped to a JND scale by calculating the difference of CSF-normalized visual energy of the distorted and the reference signal.*

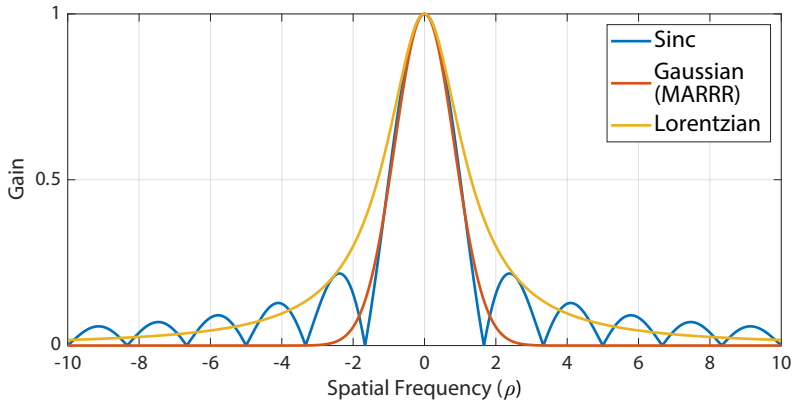


FIGURE 6.3: *Approximating sinc with monotonic functions.*

include 1 or more texels. The level of detail to sample from is then selected in such a way that the ratio of pixel-to-texel area is close to 1:1. Most modern shading languages provide a function to calculate and query this value (such as *textureQueryLod()* in GLSL). Since VRS enlarges pixel area, it results in the selection of a coarser mipmap level. Let l be the mipmap level selected when rendering at full shading resolution such that $l = 0$ is the highest (finest) resolution level. The width of the box filter at lower shading resolution can be approximated as:

$$b_\mathrm{m} = \frac{2^l}{\min(r_\mathrm{x}, r_\mathrm{y})} \quad \left[\frac{\text{texel}}{\text{pixel}}\right] . \tag{6.4}$$

The symbols used in this equation are defined in Equation 6.2. By taking the minimum of the two shading resolutions, we assume isotropic filtering, but it is trivial to extend it to support anisotropic filtering. We will use upper case symbols to denote functions in the Fourier domain. The Fourier transform of our box filter gives:

$$B_\mathrm{m}(\tau_\mathrm{u}, \tau_\mathrm{v}; b_\mathrm{m}) = \mathrm{sinc}(b_\mathrm{m}\tau_\mathrm{u}) \cdot \mathrm{sinc}(b_\mathrm{m}\tau_\mathrm{v}) . \tag{6.5}$$

However, using a sinc$(\cdot)$ function introduces non-monotonicity in the model predictions making it unsuitable for optimisation problems. To mitigate this, we approximate sinc$(\cdot)$ with a Lorentzian function $\mathcal{L}(\cdot)$ of half-width σ_m:

$$\mathcal{L}(\tau; \sigma_\mathrm{m}) = \frac{\sigma_\mathrm{m}^2}{\sigma_\mathrm{m}^2 + 4\tau^2} , \tag{6.6}$$

$$\sigma_\mathrm{m} = \frac{\pi}{2b_\mathrm{m}} . \tag{6.7}$$

We found Lorentzian function to be a good approximation of sinc$(\cdot)$ as it preserves the high frequencies similar to sinc$(\cdot)$, unlike the Gaussian approximation used by [Denes et al., 2020] (refer to Figure 6.3).

The texture at a mipmap level l is therefore affected by a low-pass filter as follows:

$$T'(\tau_\mathrm{u}, \tau_\mathrm{v}) = T(\tau_\mathrm{u}, \tau_\mathrm{v}) \cdot B_\mathrm{m}(\tau_\mathrm{u}, \tau_\mathrm{v}; b_\mathrm{m}) , \tag{6.8}$$

$$B_\mathrm{m}(\tau_\mathrm{u}, \tau_\mathrm{v}; b_\mathrm{m}) \approx \mathcal{L}(\tau_\mathrm{u}; \sigma_\mathrm{m}) \cdot \mathcal{L}(\tau_\mathrm{v}; \sigma_\mathrm{m}) . \tag{6.9}$$

Since we will now operate in screen-space, we map the mipmapped texture $T'(\tau_u, \tau_v)$ to a screen space image $I\left(\rho_x, \rho_y\right)$:

$$I\left(\rho_x, \rho_y\right) = T'(\tau_u \, 2\,\widehat{\rho}\, b_m \, , \tau_v \, 2\,\widehat{\rho}\, b_m) \quad \left[\text{cycles}/\text{deg}\right] , \tag{6.10}$$

where ρ_x and ρ_y are spatial frequencies in cpd.

VRS repeats the above texture look-up operation for all the "coarse" pixels (large VRS pixels). This is same as downsampling the filtered signal $I\left(\rho_x, \rho_y\right)$ by convolving it with a sampling function $S(\rho_x, \rho_y)$. The sampling function is a sequence of impulses at an interval of $2\,\widehat{\rho}\, r_x$ and $2\,\widehat{\rho}\, r_y$ in x and y direction, respectively. It can be represented using a Dirac comb function $\text{Ш}(\cdot)$:

$$S(\rho_x, \rho_y) = \text{Ш}\left(\frac{\rho_x}{2\,\widehat{\rho}\, r_x}\right) \text{Ш}\left(\frac{r_y}{2\,\widehat{\rho}\, r_y}\right) . \tag{6.11}$$

In our implementation of $S(\rho_x, \rho_y)$, we only considered the first two impulses to limit the discontinuities introduced by the function.

A display cannot render frequencies higher than its spatial Nyquist frequency. Hence, we can clip the high frequencies by multiplying the resultant signal with a rectangular function $R(\cdot)$:

$$I'(\rho_x, \rho_y) = \left(I\left(\rho_x, \rho_y\right) * S\left(\rho_x, \rho_y\right)\right) \cdot R\left(\rho_x, \rho_y\right) , \tag{6.12}$$

where the rectangular function is defined as:

$$R(\rho_x, \rho_y) = \Pi\left(\frac{\rho_x}{2\,\widehat{\rho}\, r_x}\right) \Pi\left(\frac{\rho_y}{2\,\widehat{\rho}\, r_y}\right) , \tag{6.13}$$

and $\Pi(\cdot)$ is a rectangular function[2].

We illustrate the above process in Figure 6.2 which analyses a moving box in the frequency domain. VRS picks coarse mipmap levels that act as a low-pass filter resulting in loss of energy. The filtered signal is then downsampled through a convolution with $S(\rho_x, \rho_y)$ that results in the creation of replicas (*aliases*) of the spectrum at multiples of the sampling interval. It is followed by multiplication with $R(\rho_x, \rho_y)$ which results in downsampling blur. If the bandwidth of the spectrum I is larger than display's effective Nyquist frequency $(\widehat{\rho}\, r_x, \widehat{\rho}\, r_y)$, higher frequencies *fold* onto lower frequencies and cause spatial aliasing.

Eye Motion Blur When the rendered image moves on the screen, our eyes undergo SPEM to track it. The motion of our eyes is continuous, however, an LCD presents the image at discrete moments in time (frames). This leads to smearing of the image on the retina and is known as *hold-type blur*. The hold-type blur can be reduced by reducing the time for which the frame is displayed (persistence). We follow the same practice as Denes et al. [2020] and approximate hold-type blur with a box filter of width:

$$b_h = \frac{\nu\, p}{2\,\widehat{\omega}} \quad [\text{deg}] , \tag{6.14}$$

where the parameters are the same in Equation 6.2. Denes et al. [2020] showed that when our eye undergoes SPEM, its tracking is imperfect. Our eyes often over- or under-estimate the object location which leads to additional blurring. The degree of blurring is proportional to the object velocity. It can also be modeled with a box filter

[2]The rectangular function is equal to 1 in $(-0.5, 0.5)$, 0.5 at -0.5 and 0.5 and 0 otherwise.

of width:

$$b_\mathrm{e} = (c_\mathrm{a}\, v + c_\mathrm{b}) \cdot p \quad [\mathrm{deg}]\,, \tag{6.15}$$

where $c_\mathrm{a} = 0.001648$ and $c_\mathrm{b} = 0.079818$ are the coefficients reported in their work. .

The blur due to the hold-type blur and imperfect eye motion is modelled the same way as the loss of resolution due to VRS and mipmapping, using Lorentzian functions in the frequency domain (approximations of sinc filter):

$$B_\mathrm{h}(\rho_\mathrm{x}, \rho_\mathrm{y}) = \mathcal{L}(\rho_\mathrm{x}; \sigma_\mathrm{h}), \tag{6.16}$$
$$B_\mathrm{e}(\rho_\mathrm{x}, \rho_\mathrm{y}) = \mathcal{L}(\rho_\mathrm{x}; \sigma_\mathrm{e}), \tag{6.17}$$

where $\sigma_\mathrm{h} = \pi/2b_\mathrm{h}$ and $\sigma_\mathrm{e} = \pi/2b_\mathrm{e}$.

Note that the blurring due to eye motion is different from the blurring introduced in the previous stages (Equation 6.9). It is directional in nature and only happens parallel to object motion, unlike blurring due to mipmapping and downsampling which is omnidirectional.

The final distorted signal reaching our retina can be calculated as

$$I''(\rho_\mathrm{x}, \rho_\mathrm{y}) = I'(\rho_\mathrm{x}, \rho_\mathrm{y}) \cdot B_h(\rho_\mathrm{x}, \rho_\mathrm{y}) \cdot B_e(\rho_\mathrm{x}, \rho_\mathrm{y})\,. \tag{6.18}$$

Contrast Sensitivity and Energy Pooling To find the sensitivity of HVS to various distortions, we can modulate the signal with the spatio-temporal contrast sensitivity function (CSF) $S(\rho, \omega, L_a, r)$. For a given spatial frequency ρ, temporal frequency ω, adaptation luminance L_a, and stimulus radius r, the CSF specifies sensitivity, i.e., the reciprocal of minimum contrast threshold visible to an average observer. For our model, we use the CSF defined by Mantiuk et al. [2021a] because it accounts for all three required dimensions together and is fitted to data up to 10 000 $\mathrm{cd/m^2}$.

Finally, we can compute the overall visual energy E_s of the distorted signal by modulating it with the CSF and integrating over the entire spectrum:

$$E_\mathrm{s}(I'') = \int_0^{\widehat{\rho}} \int_0^{\widehat{\rho}} \left| I''(\rho_\mathrm{x}, \rho_\mathrm{y})\, S\left(\sqrt{\rho_\mathrm{x}^2 + \rho_\mathrm{y}^2}, \omega, L_a, r\right)\right|^\beta d\rho_\mathrm{x} d\rho_\mathrm{y}, \tag{6.19}$$

where ω is the temporal frequency, assumed to be 0, β is the power parameter of the model and is fitted to the psychophysical data in Section 6.3.4. We set the temporal frequency ω to 0 as we consider the stimulus that is stabilized on the retina and includes blur due to motion. We set the stimulus radius r to $\frac{1}{5}$ (the reciprocal of peak frequency). The adaptation luminance L_a is approximated as the mean luminance of the considered image part (VRS block). The CSF-normalised energy E_s can be linearly mapped to a JND scale as shown by Denes et al. [2020].

Given a reference signal with no distortion

$$I^\mathrm{ref}(\rho_\mathrm{x}, \rho_\mathrm{y}) = T(2\widehat{\rho}\,\tau_\mathrm{u}\,, 2\widehat{\rho}\,\tau_\mathrm{v}) \quad [\mathrm{cycles/deg}]\,, \tag{6.20}$$

we define the total spatial distortion (d_s in Equation 6.1) as the visual energy difference[3] between the reference signal and the distorted signal:

$$d_\mathrm{s} = E_\mathrm{s}(I^\mathrm{ref}) - E_\mathrm{s}(I''). \tag{6.21}$$

[3]Different from Watson and Ahumada's [2011] visible contrast energy (ViCE). We found the above formulation to be more robust to numerical inaccuracies.

6.3.2 Temporal Distortion Model

When a continuous motion signal is sampled and displayed at discrete time intervals, the motion appears to be *jerky* or *juddery*. As discussed in Section 2.1.3, the amount of judder is known to increase with velocity and decrease with increasing refresh rate. This phenomenon can be explained by analyzing the signal in the frequency domain [Watson, 2013].

We model the temporal judder (non-smooth motion) similarly as in work by Denes et al. [2020] but in a content-adaptive manner. Temporal sampling creates replicas (aliases) of the signal in the frequency domain, each separated by the refresh rate of the display, $2\,\widehat{\omega}$ (see Figure 6.4). If the temporal bandwidth of the signal exceeds the Nyquist frequency of the display ($\widehat{\omega}$), the replica will overlap with the original signal and distort it. The temporal frequency of the first replica is $2\,\widehat{\omega}$ and the spatial frequency can be calculated as:

$$\rho_\mathrm{A} = \frac{2\widehat{\omega}}{\nu} \quad \left[\frac{\text{cycles}}{\text{deg}}\right] . \tag{6.22}$$

We can safely ignore the y spatial frequency as we had assumed the velocity in y to be 0 and so no temporal distortion. The total temporal distortion (d_t in Equation 6.1) can then be approximated as the CSF-normalised visual energy in the region of the replicated spectrum that is less than the spatial Nyquist frequency of the display ($\widehat{\rho}$):

$$d_\mathrm{t} = \int_0^{\widehat{\rho}} \int_0^{\widehat{\rho}} \left| I'(\rho_\mathrm{x} + \rho_\mathrm{A}, \rho_\mathrm{y})\, S\left(\sqrt{\rho_\mathrm{x}^2 + \rho_\mathrm{y}^2}, 2\widehat{\omega}, L_\mathrm{a}, r\right)\right|^\beta d\rho_\mathrm{x}\, d\rho_\mathrm{y}\,, \tag{6.23}$$

Similar to Equation 6.19, we approximate adaptation luminance L_a as the mean luminance of the considered image part (VRS block) and set the stimulus radius r to $\frac{1}{5}$.

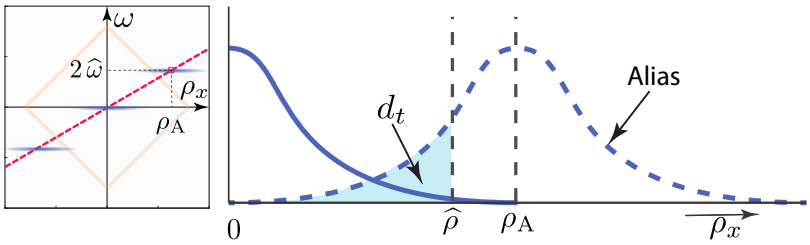


FIGURE 6.4: *Temporal distortion of the signal due to low refresh rate.*

6.3.3 Experiment 1: VRS Motion Quality Measurement

We found existing datasets to be insufficient to calibrate and test our metric. This is because the metric accounts for new factors, such as shading rate and persistence, which have not been well studied before. For that reason, we conducted a motion quality experiment, which was designed to measure the effect of those less explored factors. This section provides detailed description of the experiment.

Setup To study the effect of shading rate reduction on motion quality, observers were shown side-by-side two animations at different shading rates and textures on two separate displays. The experiment was divided into three blocks, each with

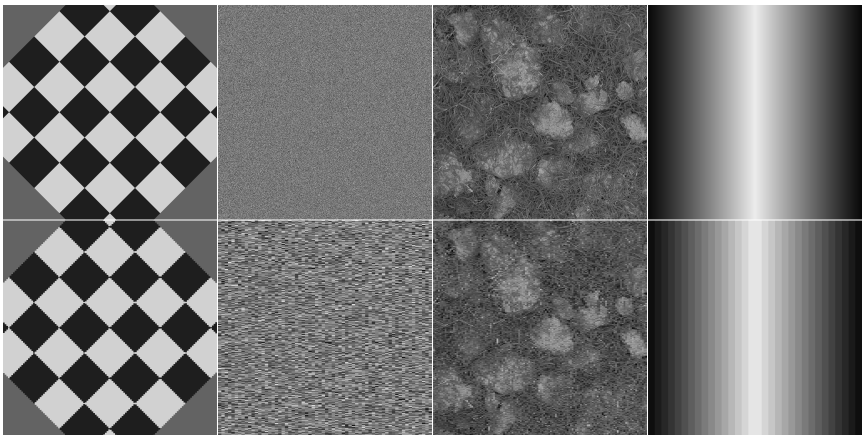
TABLE 6.1: *Range of conditions tested in each block. The same 4 textures (Checkerboard, Noise, Grass, and Gradient) processed to the correct luminance and contrast were used for all blocks.*

	Refresh Rate (Hz)	Persis-tence	Velocity (deg/s)	Resolution (ppd)	Luminance (cd/m^2)	Michelson Contrast
Block-PC	144	1	0, 10, 30	50	150	.7, .8, .9
Block-Mobile	60	1	3, 10, 20	90	75	.7, .8, .9
Block-VR	144	0.05	10, 45, 75	13	2.5	.7, .8, .9

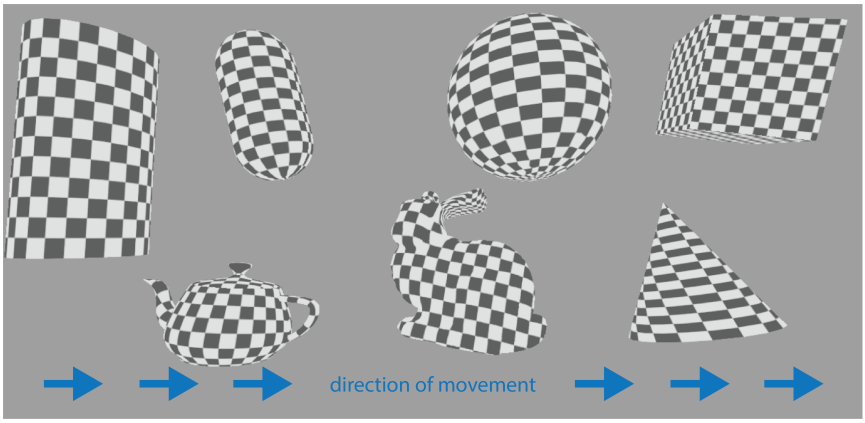
a different configuration, simulating different display devices (Table 6.1). For the first two blocks, we used a pair of ASUS ROG Swift PG279Q 27" WQHD displays, as they were capable of showing a wide range of refresh rates and luminance. For the last block, which studied low persistence, we simulated two virtual displays in a VR headset (Valve Index). We were unable to use the low persistence mode (ULMB) provided by our ASUS display as it led to excessive ghosting caused by strobe cross-talk Rejhon, 2017. The ASUS displays were calibrated with JETI Specbos 1211-2 spectroradiometer to achieve an accurate luminance reproduction and placed side-by-side to facilitate the comparison of horizontal motion. The experiment was designed in Unity3D and we used VRS offered by Nvidia RTX 2080Ti for shading rate reduction. To prevent observer fatigue, we ensured that the experiment time did not exceed 30 minutes.

Stimuli Based on the previous work detailed in Section 2.1.3, we identified the following factors to have the highest effect on the quality of motion: refresh rate, persistence, resolution, luminance, contrast, object velocity, and shading rate. Due to the high dimensionality of this problem, we restricted our experiment to three blocks according to what we believe to be the typical rendering configurations of our target application, namely, PC, mobile, and VR real-time rendering. The exact details of the range of each tested dimension can be found in Table 6.1. The resolution was varied by changing the viewing distance or scaling the viewport. For content, we used textured 3D models of simple shapes (cube, cone, sphere, cylinder, and capsule) and two popular, more complex models (Stanford bunny and Utah teapot), all shown in Figure 6.5b. We used 4 textures: checkerboard, linear-gradient, salt-and-pepper noise, and grass, shown in Figure 6.5a. The textures and contrast were chosen to bring out the most common spatial and temporal artifacts associated with shading rate reduction and to include a wide range of variations in spatial frequency content. The textures were pre-processed to produce the selected luminance and contrast levels for each block. 2× supersampling anti-aliasing (SSAA) was used only in VR (Config3) to compensate for low PPD and to reduce the visibility of the artifacts. All the models moved horizontally at the same velocity. The velocities in each block were selected to be representative of common velocities in real-time graphics on respective displays. A small rotational velocity (<2 deg/s) was added to each model randomly so that all parts of the model were visible over the experiment. Since there was no vertical motion in our stimulus, we used the only possible combination of available shading rates that would keep the vertical rate constant while varying the horizontal shading rate between $\{1 \times 2, 2 \times 2, 4 \times 2\}$ (a shading rate of $\times$N means $1/N$ shading resolution). The experiment was divided into 3 separate sessions, one for each block.

Procedure 13 participants aged 21–40, 4 females and 9 males with normal or corrected-to-normal vision participated in the experiment. The experiment used



(A) *Textures used in the experiment (top row) and their counterparts rendered using a lower shading rate (bottom row). The VRS was simulated and exaggerated using nearest-neighbor downsampling.*



(B) *Horizontally moving and slowly rotating 3D models, rendered with the checkerboard texture, that were shown in the experiment.*

FIGURE 6.5: *Textures and objects used in the experiment.*

a pairwise comparison protocol for quality assessment. In each trial, the participants were presented with two moving stimuli at different shading rates and potentially different textures. They had an option to see the reference stimuli, rendered at the original resolution (VRS 1×1), by pressing and holding the space bar. A short blank of 0.5 s was shown before switching between test and reference animations so that the participants could not use temporal flicker to detect the presence of artifacts. Participants were then asked to pick the stimulus closer to its respective reference (stimulus with fewer artifacts). Before the experiment, all participants were briefed in a short training session where we highlighted the motion artifacts caused by VRS. Since we were limited to a relatively small number of participants due to COVID-19 restrictions, we collected more data from each participant to reduce confidence intervals. Each participant performed 120 comparisons for each of the three blocks (4680 comparisons across all participants). The order of comparisons was determined using an active sampling method [Mikhailiuk et al., 2020] to maximize the information gained through each trial.

Results The pairwise comparison results from this experiment were mapped to a Just-Noticeable-Difference (JND) scale based on Thurstone Case V assumptions [Perez-Ortiz and Mantiuk, 2017]. A quality difference of 1 JND means that 75% of

the population will pick one condition over another. Since the original formulation of Thurstone Case V assumption does not allow for the computation of confidence intervals, bootstrapping was used to compute them. The results are reported in Figure 6.6a. The shading rate of 1×2 was assigned as the 0 JND condition and other data points are presented relative to it. The shapes of the plots are consistent with previous studies with quality impairment (ΔQ) due to reduced shading resolution decreasing with increasing velocity. Though this effect is not as prominent in VR. This can be explained by hold-type blur increasing with velocity and masking the spatio-temporal aliasing caused by reduced shading rate. Since VR has minimal hold-type blur because of its low-persistence, the VRS artifacts continue to be visible even at high velocities. Another important observation is that quality degrades non-linearly with shading resolution and the slope of this curve varies widely between different textures; with artifacts in gradient rarely visible and artifacts in noise almost always visible. From these results, we confirm that the motion quality depends on a wide range of display and content-related factors.

6.3.4 Model Calibration

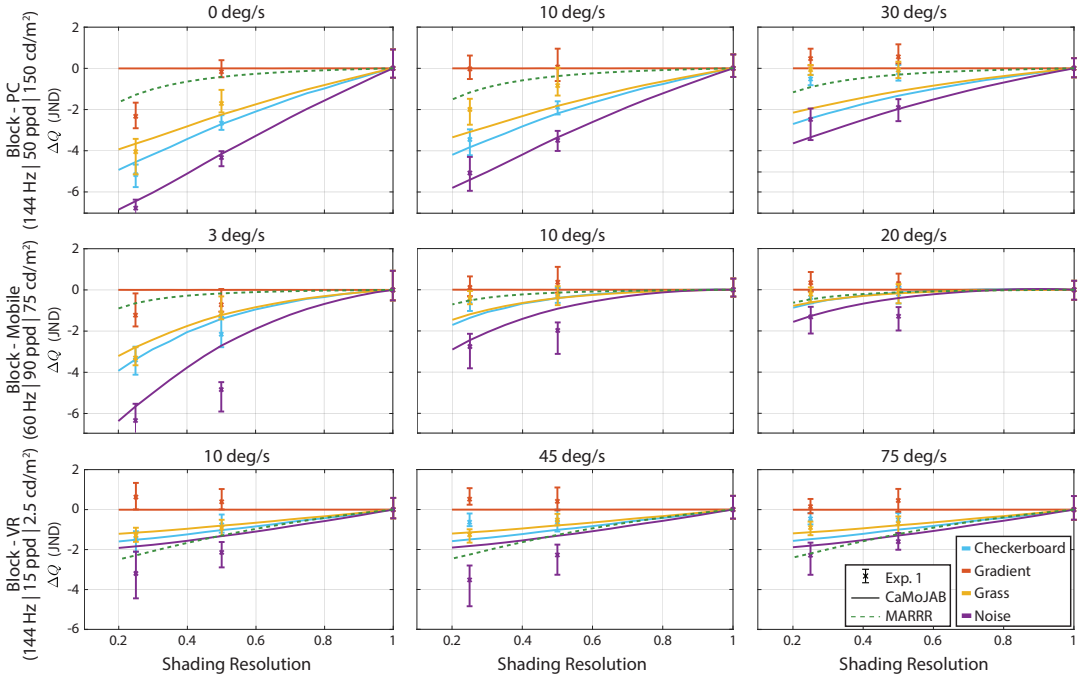
To determine the free parameter β and the relative weights of the spatial and temporal distortions w_s and w_t, we use the psychophysical data from our Experiment 1 (Section 6.3.3) in addition to Experiment 1 and 3 from [Denes et al., 2020]. The experiment by Denes et al. [2020] measured supra-threshold motion quality through moving checkerboards at different refresh rates and velocities on a full persistence LCD and no-reference pairwise comparisons. The two datasets use the same JND scale and together include a wide range of display settings and all three artifacts, i.e., motion blur, downsampling artifacts, and judder (refer to Table 2.1 for the exact range of factors tested). The values of the parameters are determined by minimizing the root-mean-squared error between the model predictions and the results of the experiments. The fitted parameter values (RMSE = 0.53) are reported in Table 6.2 and the results are plotted in Figure 6.6a and Figure 6.6b. CaMoJAB correctly predicts the reduction in quality differences with increasing velocity and successfully captures the relationship between different textures across all three configurations.

TABLE 6.2: *Model parameters*

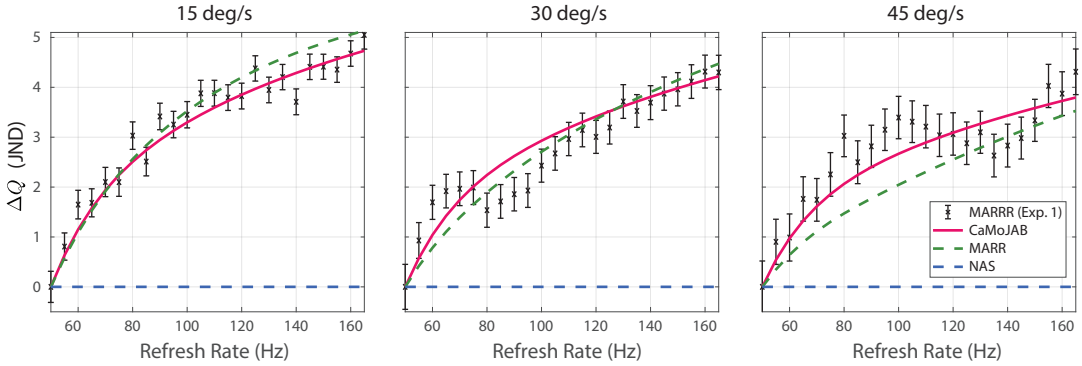
w_s	w_t	β
3.69	2.9	0.36

6.3.5 CaMoJAB Results and Comparison

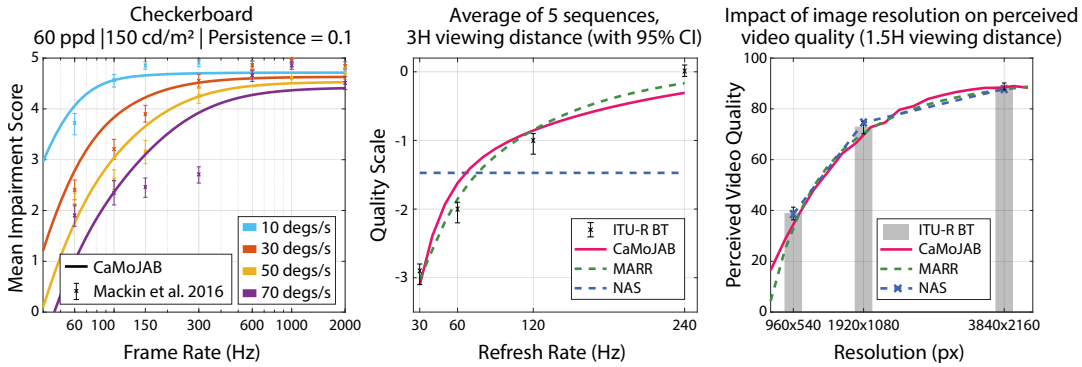
We provide a visualization of CaMoJAB predictions for the moving grass texture at different velocities, refresh rates, and shading resolutions in Figure 6.7. In the first plot, it predicts that as the velocity (in $^{deg}/_{s}$) increases, the difference in quality between high and low shading resolution decreases. This is consistent with our observations in Experiment 1 (Section 6.3.3). In the second plot, it predicts that at lower velocities, observers prefer higher shading resolution over refresh rate and the trend reverses as the velocity increases. This is consistent with the findings of Denes et al. [2020]. These model predictions hold high practical significance as they can be used to drive real-time rendering on performance adaptive hardware such as variable refresh rate (VRR) displays or VRR compatible cloud gaming [Colenbrander,



(A) *Experiment 1 results and CaMoJAB predictions scaled to JND units.*



(B) *CaMoJAB predictions on [Denes et al., 2020]'s Experiment 1.*



(C) *CaMoJAB's validation on [Mackin et al., 2016] and ITU-R [Series, 2020]. It generalizes well to complex stimuli such as natural videos and real scenes.*

FIGURE 6.6: *Experiment 1 results and CaMoJAB predictions. CaMoJAB was calibrated together on all the results reported in (a) and (b) and is capable of capturing the correct trends. MARRR being a content-independent model, is unable to capture the difference in quality due to texture in (a). NAS does not account for display factors and thus fails to predict the change in quality due to refresh rate in (b).*

2021], and VRS GPUs. We describe and implement one such strategy in a complex motion setting in Section 6.4. Finally, our model also provides an opportunity to study the interplay between luminance and persistence (Figure 6.8) for rendering on the upcoming variable persistence-variable refresh rate displays [Verbeure et al., 2017; Hekstra et al., 2008].

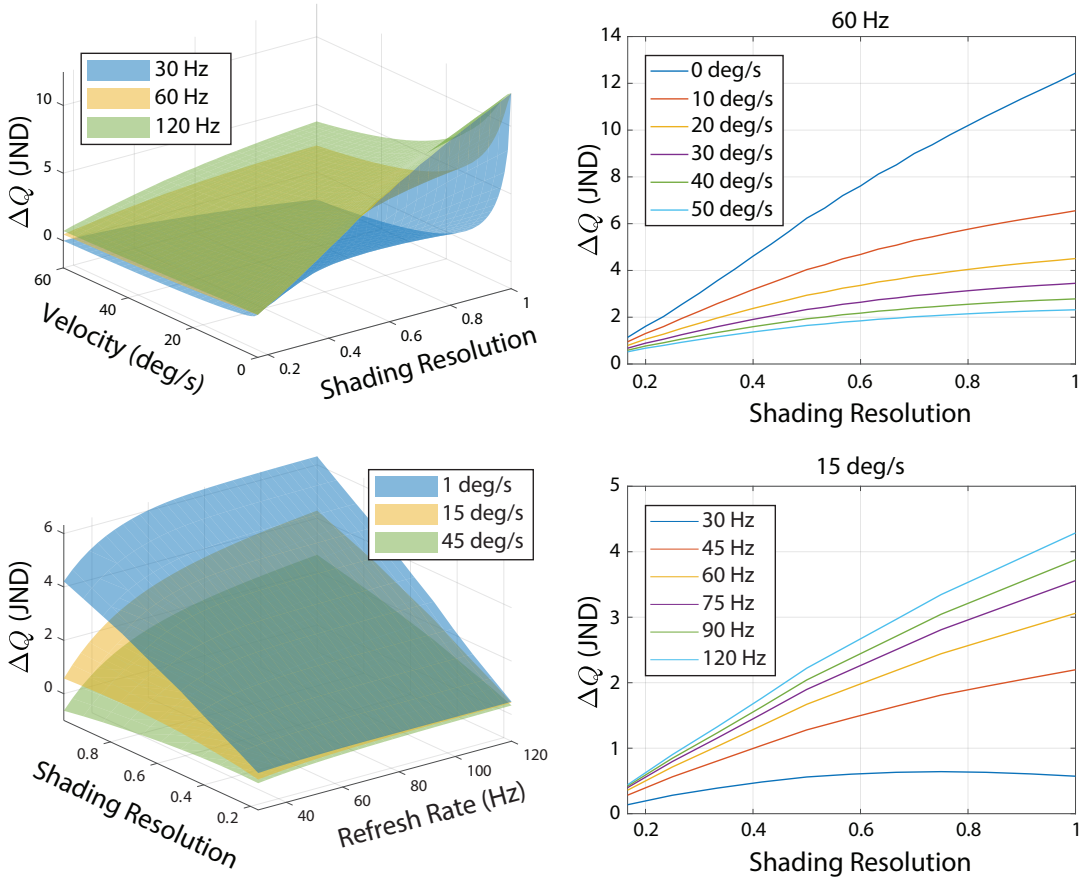


FIGURE 6.7: *CaMoJAB predictions for a moving grass texture on a* $150cd/m^2$ *and full persistence 60 ppd display. Although we plot shading resolution as a continuous variable, in practice, only a small set of resolutions are available.*

For a comparative study, we picked two state-of-the-art models proposed by Denes et al. [2020] and Yang et al. [2019]. It should be noted that neither of these models (or any other model to our knowledge) is designed to predict our kind of data. Solution form Denes et al. [2020] is a perceptually based model derived through extensive psychophysical experiments, however, it is content-independent. The method designed by Yang et al. [2019], though content dependent, does not take display and viewing conditions into consideration. Furthermore, it is designed for a specific task to give quality scores for only full, half, and quarter shading resolution and not a continuous scale. For a fair comparison, both models were linearly fitted to our data. As can be seen from the quantitative comparison in Table 6.3, our model predicts the trends significantly better than the other two models across all datasets.

6.3.6 CaMoJAB Validation

To further validate and verify the generalizability of our model, we test our model's performance on motion quality datasets from Mackin et al. [2016] and ITU-R recommendations [Series, 2020]. The experiment by Mackin et al. [2016] used mean opinion

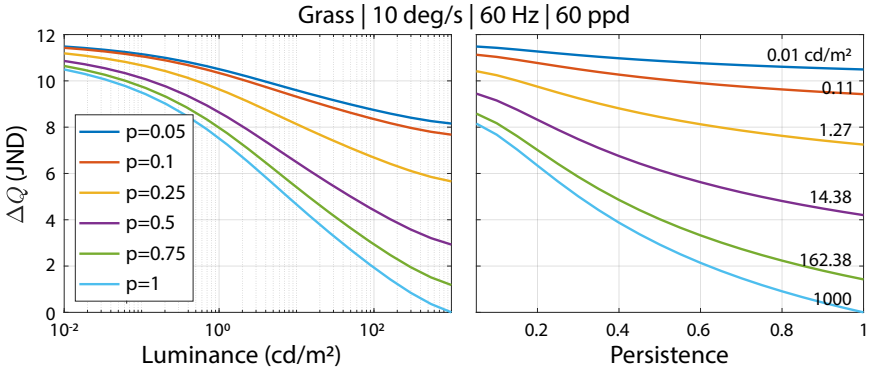


FIGURE 6.8: *Luminance-persistence trade-off for a 60 Hz animation on a 60 ppd display. The motion artifacts in such animation are more noticeable at high luminance and for longer display persistence.*

TABLE 6.3: *MARRR, NAS, and CaMoJAB (Ours) performance on motion quality datasets. All models were linearly fitted to each dataset. Numbers denote the RMSE values. We report RMSE corresponding to a moving checkerboard for the ITU dataset as they did not report the stimulus used.*

	Block-PC	Block-Mobile	Block-VR	MARRR Exp. 1	MARRR Exp. 3	Mackin et al. [2016]	ITU-R FPS [Series, 2020]	ITU-R PPD [Series, 2020]
CaMoJAB (ours)	**0.74**	**0.57**	**0.53**	**0.38**	**0.03**	**0.42**	0.28	0.69
MARRR	1.82	1.55	1.14	0.55	0.04	0.55	**0.15**	**0.25**
NAS	1.41	1.1	1.23	3.12	0.21	1.04	1.09	1.19

score methodology to study motion quality of real-continuous motion of rotating lines on a low persistence display (simulated via strobing lights on a printed paper). The experiments in ITU-R [Series, 2020] use natural videos as stimuli and their results form the basis of presently in-force ITU recommendations.

We linearly scaled our model to fit their data and bring the predictions to the same perceptual scale (Figure 6.6c). We report the goodness of fit (RMSE) for a moving checkerboard in Table 6.3. The results show that our model also generalizes well to more complex stimuli.

6.4 Adaptive Local Shading and Refresh Rate

Modern graphics engines require dynamic quality control that would adapt rendering parameters (resolution, refresh rate, level of detail, and others) to the available resources (GPU frame-budget, power). This becomes especially important for mobile devices with high-resolution and refresh-rate displays, but also in time-sharing cloud rendering, used for streaming of games. Current solutions used in popular game engines typically involve dynamic resolution scaling of the entire frame to meet these constraints [Unreal, 2021; Unity, 2021]. This is a sub-optimal approach, as better quality can be obtained by (a) adaptively selecting the best combination of resolution and refresh-rate [Denes et al., 2020] and (b) adjusting the resolution locally, using VRS. In this section we describe our Adaptive Local Shading and Refresh Rate (**ALSaRR**) algorithm that uses our motion quality metric to determine the optimal distribution of shading rate and refresh rate under a given bandwidth constraint.

An overview of our ALSaRR can be found in Figure 6.9. It takes as input the allowed bandwidth B in pixels per second and several auxiliary buffers rendered at the resolution of the VRS map: motion vectors, texture IDs, mipmap levels and luminance. The pooled motion vectors and the bandwidth B are used to determine

the best frame rate for the current frame. Then, the maps are fed to the CaMoJAB to compute for each VRS block the ratios of quality to bandwidth for all possible shading rates. The ratios are then used by a greedy knapsack solver to find the distribution of shading rates that maximize the quality at the given budget. In the following subsections, we describe each step in detail and how they can be implemented in a real-time game engine.

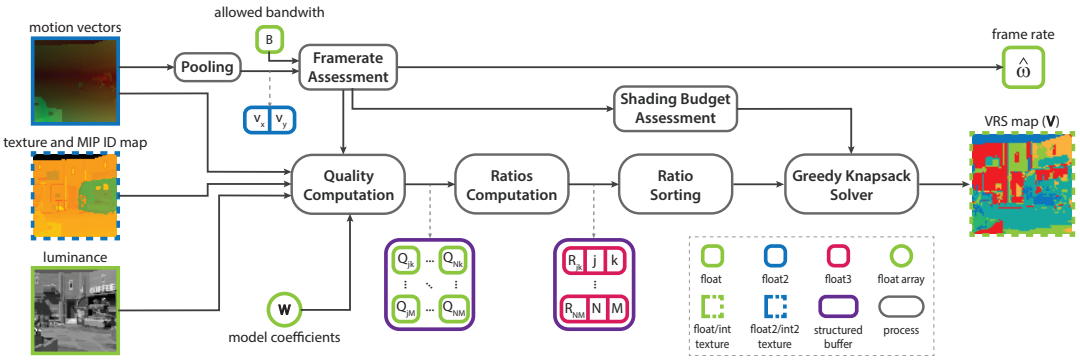


FIGURE 6.9: *The processing diagram for Adaptive Local Shading and Refresh Rate (ALSaRR). The method uses motion vectors, luminance, and precomputed polynomial coefficients to predict the optimal refresh rate and distribution of the shading rates given allowed bandwidth.*

6.4.1 Auxiliary Buffers

Our quality metric requires the knowledge of texture, the mapping of texels to fragments, luminance, and velocity of the pixels rendered on the screen. For that purpose, we render into a G-buffer: texture IDs, MIP map level, motion vectors, and luminance at the resolution of the VRS map (width/16 $\times$ height/16 pixels). As rendering luminance would require additional shading, which we want to avoid, we use the previously rendered frame (without reprojection) to approximate luminance. The previous frame is first subsampled, then converted to grayscale luma, from luma to linear absolute luminance values, and scaled according to the display peak luminance.

6.4.2 Optimal Refresh Rate

Adaptive-sync displays allows us to chose arbitrary refresh rates, which let us either improve smoothness of motion or spatial resolution, depending on the velocity of motion. Similar to MARRR, we express this as an optimization problem:

$$\begin{aligned} &\arg\max_{\widehat{\omega}} q\left(I, v_{\text{frame}}, \widehat{\omega}, p, \widehat{\rho}, r_{\text{x}}, r_{\text{y}}\right) \\ &\text{s.t.} \quad 2\,\widehat{\omega}\,R_{\text{x}}\,r_{\text{x}}\,R_{\text{y}}\,r_{\text{y}} = B \;\wedge r_{\text{x}} = r_{\text{y}} \wedge\; 2\,\widehat{\omega} \in \mathbb{Z}^{+} \;\wedge\; 2\,\widehat{\omega} \leq \omega_{\max}, \end{aligned} \tag{6.24}$$

where q is the quality function defined in Equation 6.1, R_x and R_y are the horizontal and vertical display resolution in pixels, $\omega_{\max}$ is the maximum refresh rate of the display and B is the fixed budget in pixels-per-second. As the refresh rate is global for the frame, we take the entire screen-space image (I) as a proxy for the image being tracked and pool the motion buffer to compute the average velocity of the frame (v_{frame}) in $^{\text{deg}}/_{\text{s}}$. The rest of the parameters are the same as in Equation 6.2.

Once we have the optimal refresh rate $2\widehat{\omega}$, we can calculate the maximum allowable shading budget of the given frame as

$$b_{\text{frame}} = \frac{B}{2\widehat{\omega}} \quad [pixels]. \tag{6.25}$$

Since solving the optimization for every frame is too expensive for real-time rendering, we precompute the best combinations using the first frame and store them as a 1D LUT of velocity vs. refresh rate (and resolution).

6.4.3 Optimal Shading Rates

In this section, we describe how to obtain an optimal VRS map under a limited shading budget b_{frame}.

Problem Formulation A screen-space image can be divided into N 16×16 pixel VRS tiles ($\boldsymbol{\mathcal{T}}_j$, $j = 1, \ldots, N$). Each tile $\boldsymbol{\mathcal{T}}_j$ has a bandwidth ($\boldsymbol{\mathcal{B}}_{jk}$) and a quality ($\boldsymbol{\mathcal{Q}}_{jk}$) corresponding to k^{th} shading rate from the set of all available shading rates $\boldsymbol{\mathcal{R}}$.

$$\boldsymbol{\mathcal{R}} = \{4 \times 4, 4 \times 2, \ldots, 1 \times 1\} \tag{6.26}$$

such that for $\boldsymbol{\mathcal{R}}_k = m \times n$. The bandwidth can be calculated as

$$\boldsymbol{\mathcal{B}}_{jk} = \frac{256}{m \cdot n} \quad [\text{pixels}]. \tag{6.27}$$

Since our motion quality model assumes unidirectional motion, we calculate the quality in x and y direction separately using the quality function from Equation 6.1, so that

$$\begin{aligned} q_x(\boldsymbol{\mathcal{T}}_j, \boldsymbol{D}, \boldsymbol{\mathcal{R}}_k) &= q\left(\boldsymbol{\mathcal{T}}_j, v_j^{\text{x}}, \widehat{\omega}, p, \widehat{\rho}, \frac{1}{m}, \frac{1}{n}\right) \quad [\text{JND}], \\ q_y(\boldsymbol{\mathcal{T}}_j, \boldsymbol{D}, \boldsymbol{\mathcal{R}}_k) &= q\left(\boldsymbol{\mathcal{T}}_j^{\top}, v_j^{\text{y}}, \widehat{\omega}, p, \widehat{\rho}, \frac{1}{n}, \frac{1}{m}\right) \quad [\text{JND}], \end{aligned} \tag{6.28}$$

where v_j^{x} and v_j^{y} are the minimum velocity in the tile j along x or y (the most conservative assumption). The rest of the parameters are the same as in Equation 6.2. We define the total quality of the tile as the mean of q_x and q_y

$$\boldsymbol{\mathcal{Q}}_{jk} = \left(\frac{q_x(\cdot) + q_y(\cdot)}{2}\right) \quad [\text{JND}]. \tag{6.29}$$

Finally, given the maximum allowable bandwidth per frame b_{frame}, we need to select the shading rate k for each tile $\boldsymbol{\mathcal{T}}_j$ to find

$$\arg\max_{\boldsymbol{\mathcal{B}}_{jk}} \sum_{j=1}^{N} \boldsymbol{\mathcal{Q}}_{jk} \quad \text{subject to} \quad \sum_{j=1}^{N} \boldsymbol{\mathcal{B}}_{jk} < b_{\text{frame}} \tag{6.30}$$

Greedy Solution The optimal solution of the above knapsack problem can be found using dynamic programming but the algorithm has the time complexity of $\Theta(N \cdot b_{\text{frame}} \cdot |\boldsymbol{\mathcal{R}}|)$ and the space complexity of $\Theta(N \cdot b_{\text{frame}})$, making it unsuitable for real-time rendering (alone $N = 8100$ for the FullHD resolution). Instead, we solve the problem using an approximated greedy solver [Kellerer et al., 2004, Section 2.1], which has the time complexity of $\Theta(N \cdot |\boldsymbol{\mathcal{R}}|)$ and the space complexity of $\Theta(N \cdot |\boldsymbol{\mathcal{R}}|)$.

The algorithm finds near-optimal allocation of shading rates based on the quality predictions. Its pseudocode can be found in Algorithm 1.

Let $\boldsymbol{V}$ be a vector of N elements storing the current shading rate for each tile j in $\boldsymbol{V}_j$. We initialize $\boldsymbol{V}$ with the index of lowest bandwidth shading rate, $\boldsymbol{V}_j = 1 \; \forall \; j \in [1, N]$, and subtract the cost of rendering at that rate from the current frame bandwidth budget, b. Next, we calculate all possible ratios of quality to bandwidth for each tile, $\boldsymbol{R}_{jk} = \mathcal{Q}_{jk}/\boldsymbol{B}_{jk}$, and sort them in non-increasing order. The first element in the sorted list will give the tile and shading rate that increases the quality at the least cost (bandwidth). Next, we greedily pick (j, k) pairs from the head of the sorted list and update the current frame budget, $b = b - \boldsymbol{B}_{jk} + \boldsymbol{B}_{j\boldsymbol{V}_j}$ and shading rate of j^{th} tile, $\boldsymbol{V}_j = k$. The steps are repeated until the remaining budget becomes 0.

Result: near-optimal shading rate of each tile
$b \longleftarrow b_{\text{frame}}$

for j = *1 to* N ; // For each tile
do
  $\boldsymbol{V}_j \longleftarrow 1$; // Shading rate 4×4
  $b \longleftarrow b$ - $\boldsymbol{B}_{j1}$; // Update remaining bandwidth
  for k = *2 to* $|\boldsymbol{\mathcal{R}}|$; // For each shading rate
  do
    $\boldsymbol{R}_{jk} \longleftarrow \frac{\mathcal{Q}_{jk}}{\boldsymbol{B}_{jk}}$; // Calculate ratios
  end
end
sort($\boldsymbol{R}$) ; // From the highest to the lowest
while $b > 0$ **do**
  (j,k) $\longleftarrow \boldsymbol{R}$.pop() ; // Pick the highest ratio
  $b \longleftarrow b$ - $\boldsymbol{B}_{jk}$ + $\boldsymbol{B}_{j\boldsymbol{V}_j}$; // Update the bandwidth
  $\boldsymbol{V}_j \longleftarrow$ k ; // Update the shading rate
end

Algorithm 1: Greedy solution for the modified knapsack problem.

To ensure that the near-optimal solution is sufficient, we compared the results of the greedy algorithm with the accurate solution. Although the theoretical error in the estimated near-optimal solution can be up to 50% of the optimal objective function value, we found that in practice the greedy method has the expected error of less than 1% in our tested scene while having orders of magnitude lower time and space complexity.

Real-Time Implementation To reduce the performance overhead of our method, the quality predictions of CaMoJAB (Equation 6.1) are precomputed for every texture, mipmap level, and shading rate and stored as polynomials of the form:

$$\begin{aligned} q_{i,j,k}(\nu, f, L) = (W_0 + W_1\nu + W_2 f + W_3\nu^2 + W_4\nu f + W_5 f^2) \\ \cdot (W_6 L^3 + W_7 L^2 + W_8 L + W_9), \end{aligned} \tag{6.31}$$

where i is the texture index, j is the MIP map level, k is the shading rate, ν is velocity, f is the refresh rate of the current frame ($2\,\widehat{\omega}$), L is the logarithmic luminance of the

tile, and $W_{0,\ldots,9}$ are the coefficients of the polynomial, stored separately for each (i, j, k) triplet.

Our real-time implementation follows Algorithm 1 and computes the ratios of quality to bandwidth for each tile and each possible shading rate. Those ratios are sorted in a non-increasing order (the best quality-to-shading rate first) using a bitonic sort, as this algorithm can utilize the parallelism of the GPU.

6.4.4 The Velocity of Eye Motion Without Eye Tracking

In a typical animation, multiple objects can move on the screen in different directions with different velocity and the eye could follow any of them. For that reason, MARRR method required an eye tracker to determine the actual velocity of the eye motion relative to the screen. However, because we control the shading rate locally, we can make a conservative assumption that the eyes could follow each of 16×16 tiles. This lets us use different velocity of eye motion for each tile instead of a global velocity per frame, as done in MARRR. Because the distortions are the most visible when the object is followed, we get the worst-case estimate of the degradation of quality due to the reduced shading rate. This gives us an important practical advantage over MARRR as eye tracking is still rare in most applications of real-time graphics.

6.4.5 Implementation Details

To test our method on realistic scenes of sufficient complexity, we implemented a native plugin for Unity. The plugin modifies the default forward rendering pipeline by allowing for a custom VRS map. VRS is enabled only for the fragment shader stage and is disabled for the post-processing pass, as that would introduce reduced shading at the pixel level and shade fragments across the edges.

Performance We measured the performance of ALSaRR on a GeForce RTX 2080Ti graphics card using built-in Unity profiler. Our implementation utilizes parallelism of GPU and requires, on average, 1.3 ms to process the frame of resolution 1920 × 1080 pixels. It also requires approximately additional 100 MBs of VRAM to store intermediate data. The processing time is independent of the rendered content and does not include preprocessing (fitting of the polynomials, Equation 6.31). We did not attempt to implement the method on a mobile GPU but we expect the efficient implementation on mobile architectures may require further performance optimizations, such as computing VRS maps at lower resolution and upsampling them.

6.4.6 Experiment 2: ALSaRR for Real-Time Rendering

In this section, we report on a pairwise-comparison preference experiment, which compared our ALSaRR method against constant-refresh-rate-resolution and MARRR. The experiment simulated video game scenarios, all implemented in Unity 3D and rendered with one of the compared methods.

Stimuli We adapted 3 scenes from Unity Asset store (Figure 6.10). For every scene, we prepared 4 different camera motion scenarios. One of them was mimicking a first- or third-person game, allowing the participants to control the camera with the mouse and keyboard. The rest of them contained predefined motion paths, each one with a different velocity of camera motion. Each scene was rendered using one of the

three compared methods: ALSaRR, constant-refresh-rate-resolution, and MARRR. For the constant-refresh-rate-resolution method, we used several combinations of shading and refresh rates, each with a fixed budget of 6.25% or 12.5% pixels of the full bandwidth (1920×1080@120 Hz). We used a lower resolution than the native resolution of the display because we noted in the pilot experiments that participants have difficulty distinguishing between the conditions at higher resolutions (on a 27" monitor seen from 75 cm).

MARRR method originally required an eye tracker to determine the velocity of eye motion (relative to the screen). However, for closer comparison, we made the same simplifying assumption as we did for the selection of refresh rate in our method and estimated the eye motion velocity as the average velocity within the entire frame (pooled from the motion buffer).

The rendered scenes were shown on the same monitor as used in Experiment 1 (Section 6.3.3).

Procedure The experiment consisted of pairwise comparisons between the three compared rendering methods. In each trial, one of the conditions was our method and the other condition was either a constant-resolution-refresh-rate or MARRR. The order of the scenes and the compared methods was randomized. The camera motion scenario was randomly selected for each trial as a full factorial design was infeasible within a reasonable time budget.

The participants could press the *space* bar to switch between two compared rendering methods. They had to choose the rendering method with overall better visual quality by pressing the *Enter* key while the preferred rendering mode was active. The question shown to the observers was: *"Which of the two methods has higher visual quality (i.e. smooth motion and sharp image)?"*. Each observer performed 108 comparisons (3 scenes × 6 budgets/rendering methods × 6 repetitions). To prevent fatigue, the experiment was split into 2 sessions lasting 30 minutes each.

Participants and Procedure 12 participants (aged 24-41) with normal or corrected-to-normal vision participated in the experiment. They were compensated for their time.

Results The results of the validation experiment, visualized in Figure 6.11, show a clear preference for our ALSaRR method, as compared to fixed resolution rendering and MARRR. The difference is significant across all tested conditions (binomial test with the null hypothesis of random guess). As expected, the improvement due to adaptive rendering decreases as the bandwidth increases because the differences between the conditions become less noticeable. MARRR performed better than fixed-frame-rate at a higher budget of 12.5%, but worse at 6.25%. It should be noted, however, that our fixed-rate conditions used reduced shading rate instead of bilinear upsampling used in [Denes et al., 2020]. This may indicate that reducing the shading rate may produce better visual results than upsampling from a lower resolution when the rendering budget is low. The lack of eye tracking could also negatively affect the performance of MARRR.

[4] https://assetstore.unity.com/packages/essentials/tutorial-projects/adventure-sample-game-76216

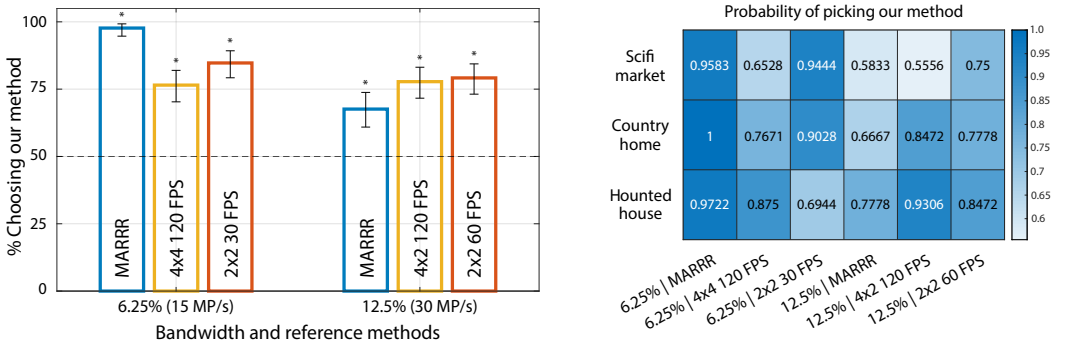
[5] https://learn.unity.com/project/john-lemon-s-haunted-jaunt-3d-beginner

[6] https://assetstore.unity.com/packages/3d/environments/urban/sally-s-country-home-33123



FIGURE 6.10: *Preview of the scenes selected for validation experiment. The SciFi Market scene [4] (top) and John Lemon's Haunted Jaunt scene [5] (bottom) have been released as learning materials by Unity. The Sally's Country House scene [6] (middle) created by Gabro Media has been acquired from Unity Asset Store.*

The per-scene results, shown in Figure 6.11b, suggest the performance gain of our method was the smallest for the “Scifi market" scene. After closer inspection, we noted that the scene contained a large number of hard shadows, which resulted in aliased edges for our method. This is because our current implementation of ALSaRR relies on the information from textures, rather than rendered fragments and, therefore, ignores local shading (with the exception of mean luminance). This problem is difficult to mitigate in real-time rendering as the decision on shading rate must be made before local shading is computed.



(A) *Results of Experiment 2, aggregated across all scenes* (B) *Results of Experiment 2 for each scene*

FIGURE 6.11: *Results of the validation experiment showing percentage of participants picking our method over MARRR and constant-refresh-rate-resolution rendering. Error bars in (a) denote 95% confidence interval.*

6.5 Limitations and Discussion

One of the important advantages of our method over foveated rendering [Tursun et al., 2019] and MARRR is that it does not require eye tracking. However, to eliminate the dependency on eye tracking, we had to make a conservative assumption that the eye can follow movement in every VRS tile. If the gaze location was known to us, we could potentially further improve the predictions and therefore allocation of the rendering budget. Our method could be also combined with foveated rendering for further gains in quality.

Our CaMoJAB metric accounts for most relevant factors of motion quality, including texture content, on-screen velocities, luminance, effective resolution, and display persistence. However, it does not model contrast masking [Mantiuk et al., 2021a], degradation of color, foveation [Tursun et al., 2019] or saliency. Those limitations are partially dictated by the scarcity of available data, and partially by the real-time requirement of our application. The complexity of contrast masking would make the metric too costly for a real-time application, and foveation would require eye tracking. Also, CaMoJAB does not account for individual variations and instead models the average observer. Modeling individual variations would require much more measurements from each participant.

In comparison to *Nvidia Adaptive Shading* (NAS) [Yang et al., 2019], CaMoJAB is calibrated in physical units (pixels-per-degree, degrees-per-second, cd/m^2) and accounts for the display characteristics that affect motion quality perception, such as refresh rate, duty-cycle, and field-of-view. NAS may need to be recalibrated for a display of different size, brightness or pixel density than the one used by the authors. CaMoJAB has also been calibrated with several datasets, collected using psychophysical procedures, rather than tuned by the authors.

Practical Considerations for ALSaRR Our ALSaRR method can control real-time rendering with an acceptable overhead because we can pre-compute the quality predictions for all textures, mipmap levels, and shading rates. The main shortcoming of this approach is that our CaMoJAB quality metric is unaware of the shading in the final image, for example, when a tile contains a shadow boundary. A few such failure cases are demonstrated in Figure 6.12 where ALSaRR assigns non-optimal shading rates due to the lack of correct frequency information. The examples shown in the figure include (1) reflections, (2) transparent objects and (3) procedural particle systems. Some game engines may not support rendering motion vectors for transparent materials further exacerbating the problem. It should be noted that ALSaRR uses diffuse textures as a proxy for object frequency information. Though this served as a good approximation for our tested scenes, it may not hold true for more complex materials. These issues can be addressed to an extent by precomputing the worst-case scenario shading and then using that for our precomputed quality functions. We could also use the approach from NAS [Yang et al., 2019] and directly analyze the frequency information by reprojecting the previous frame. This approach, however, relies on the assumption that the previous frame has been rendered with sufficient quality, and it also adds a substantial overhead of reading and processing full-resolution frames.

Our current implementation of ALSaRR operates on the rendering budget expressed in pixels per second while most applications are limited instead by the power of GPU cycles. This is not a fundamental limitation of our method, as it can work with the budget expressed in any units as long as there exists a model that can predict such units from the VRS rates.

6.6 Conclusions

There is a shift in real-time graphics from rendering with a fixed resolution and refresh rate to a more adaptive approach, in which we control spatio-temporal resolution in order to maximize the quality under a given rendering budget. VRS and adaptive refresh-rate monitors provide an opportunity for such a fine control, which becomes particularly important as the increasing resolution and refresh rate of displays places a high demand on the GPU, especially on mobile devices. Our ALSaRR method takes the advantage of the limitations of the visual system to reallocate rendering budget to the most vital part of the spatio-temporal domain. The key component of the method is our new metric, CaMoJAB, which considers how the judder, aliasing, and blur artifacts introduced by the VRS at a given refresh rate are masked by hold-type blur,

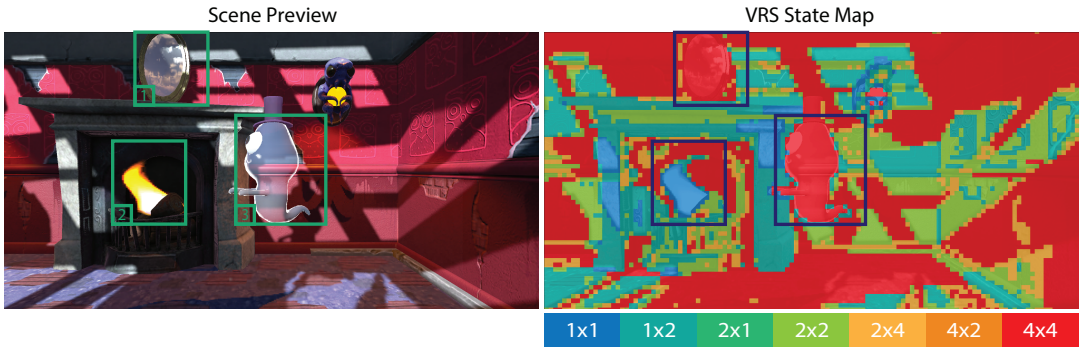


FIGURE 6.12: *Examples of fail cases: because ALSaRR relies on textures rather than shaded pixels, it assigns low shading rates to mirror reflections (1) and transparent ghost (2); high shading rate to fire particles (3).*

eye motion blur, and limited spatio-temporal sensitivity of the visual system. The metric is shown to explain multiple datasets with only a few fitted parameters.

We hope the proposed technique can improve the visual quality not only for regular adaptive-sync monitors, but also for VR headsets. Relatively high velocities of motion in those headsets have the potential to mask a portion of VRS artifacts. The adaptive choice of resolution and refresh rate could improve the quality of experience as the users would not be forced to chose one or another. Finally, the technique could reduce the rendering cost and extend battery life, especially when combined with foveated rendering.

Chapter 7

Improving Depth Perception Under Low Luminance

7.1 Introduction

There are a number of benefits of using darker displays for VR headsets. An obvious benefit is reduced power consumption, as the display itself can be responsible for half of the power usage in a stand-alone, battery-powered headset. It has been also postulated that a lower display brightness in VR/AR can reduce VR sickness, however, no evidence for that has been found so far [Vasylevska et al., 2019]. A less obvious benefit is an improved quality of the reproduced motion. Most VR displays are equipped with low persistence, in which an image is displayed at a higher intensity for a fraction of frame duration and the display remains blank for the rest of the frame. Without the stroboscopic effect introduced by such low persistence displays, we would see a significant amount of blurring caused by eye gaze moving over the image, which is stationary on the display over the duration of a frame. However, low persistence can also introduce visible flicker if the refresh rate is not high enough [Hoffman and Lee, 2019]. Lowering display luminance can reduce the visibility of such flickering without the need to increase the refresh rate [Chapiro et al., 2019b].

However, dimming a display also introduces a number of undesirable effects: it reduces the perceived contrast of an image [Barbur and Stockman, 2010; Wanat and Mantiuk, 2014], makes the image appear less colorful [Cao et al., 2008; Shin et al., 2004; Kwak et al., 2003], and affects our ability to make depth judgements based on stereoscopic depth cues. While the former two effects have been well studied and addressed in the literature, the effect of absolute luminance on our ability to see depth has received relatively less attention.

The main goals of this chapter are: (a) to measure and quantify the effect of display luminance and contrast on our ability to make depth judgements from binocular disparity depth cues, and (b) to propose an image contrast enhancement technique that can enhance depth perception on dimmed stereoscopic displays. Based on a series of psychophysical measurements (Section 7.2) on our prototype stereoscopic high dynamic range (HDR) display, we propose a model of *stereoscopic constancy* (Section 7.3), which predicts the amount of physical contrast needed to maintain the same precision of binocular disparity depth cues across the luminance range of $0.1\,\mathrm{cd/m^2}$ to $1000\,\mathrm{cd/m^2}$. The model is then used to develop a multi-scale contrast compensation method (Section 7.4) that attempts to preserve the precision of binocular depth cues at different display luminance levels. The method has been implemented in GPU shaders and it can be used in real-time applications. Then, we test our algorithm in a low-brightness VR rendering application, in which our method is both preferred and gives a better impression of depth than non-processed rendering and

existing methods (Section 7.5). Finally, we discussed gathered insights and limitations (Section 7.6) and conclude this chapter in Section 7.7.

7.2 Experiment 1: 3D Shape Perception

Binocular disparity is one of the most important depth cues [Cutting and Vishton, 1995] that is commonly employed in VR and cinematographic applications to trigger stereo 3D scene appearance. In this work, we develop a computational model of the precision of binocular disparity cues as a function of image contrast and luminance. To this end, we designed an experiment in which observers were asked to judge the angle of a 3D hinge-like shape (Figure 7.1, left), reproduced on the display using only disparity depth cues (Figure 7.1, right). Our 3D shape perception experiment was inspired by the study of Watt et al. [2005], where a similar hinge-like shape was used to examine whether focus cues have an indirect effect on depth interpretation.

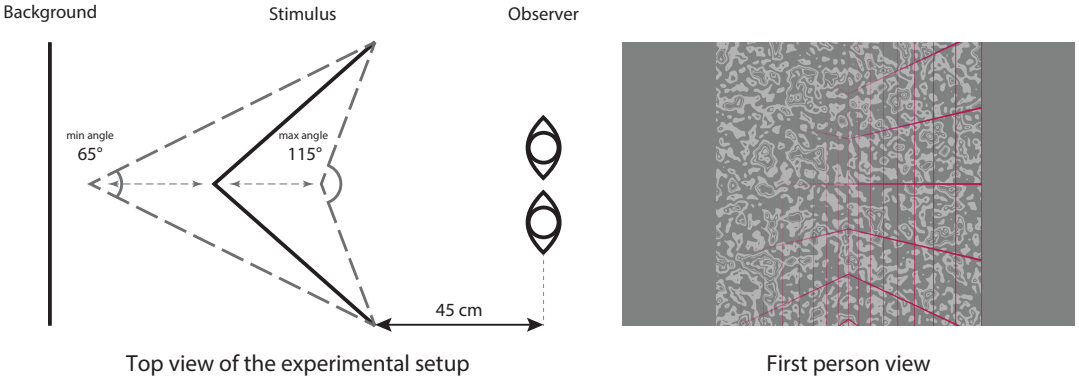


FIGURE 7.1: *Left: The stimulus used in Experiment 1. The observer is presented with a hinge-like concave shape. The angle is changed by moving the hinge part towards or away from the observer (depicted by the arrows). Right: Procedural organic pattern on a uniform background. The superimposed grid depicts the three-dimensional shape of the stimuli. Note that the superimposed grid was only added to this figure to facilitate its 3D interpretation, while originally the hinge shape was reproduced only by the disparity cue.*

Apparatus The experiment was conducted on a custom-built stereo high dynamic range (HDR) display, which allowed a single observer to view a pair of stereo images through an optical arrangement similar to the Wheatstone mirror stereoscope, as illustrated in Figure 7.2. We used a stereoscopic HDR display to explore a much larger range of luminance than that available on regular displays. The stereoscopic stimuli were shown by two projector-based 9.7" HDR displays, 2048×1536 resolution each. The peak luminance of each display was above 3,000 cd/m^2 and the black level was below 0.01 cd/m^2. Further details on the display design, control software and its colorimetric and geometric calibration can be found in [Zhong et al., 2021]. The display algorithm used spatio-temporal dithering to avoid banding artifacts caused by insufficient bit-depth. Additional information on HDR display calibration for psychophysical measurements, which were conducted with similar but monoscopic display, can be found in [Wuerger et al., 2020]. We used HDR rather than a standard display as it allowed us to reproduce both very low and very high luminance while maintaining sufficient contrast and bit-depth accuracy. The virtual images of the HDR content were placed 45 cm in front of the observers, with a resolution of 82 pixels per visual degree.

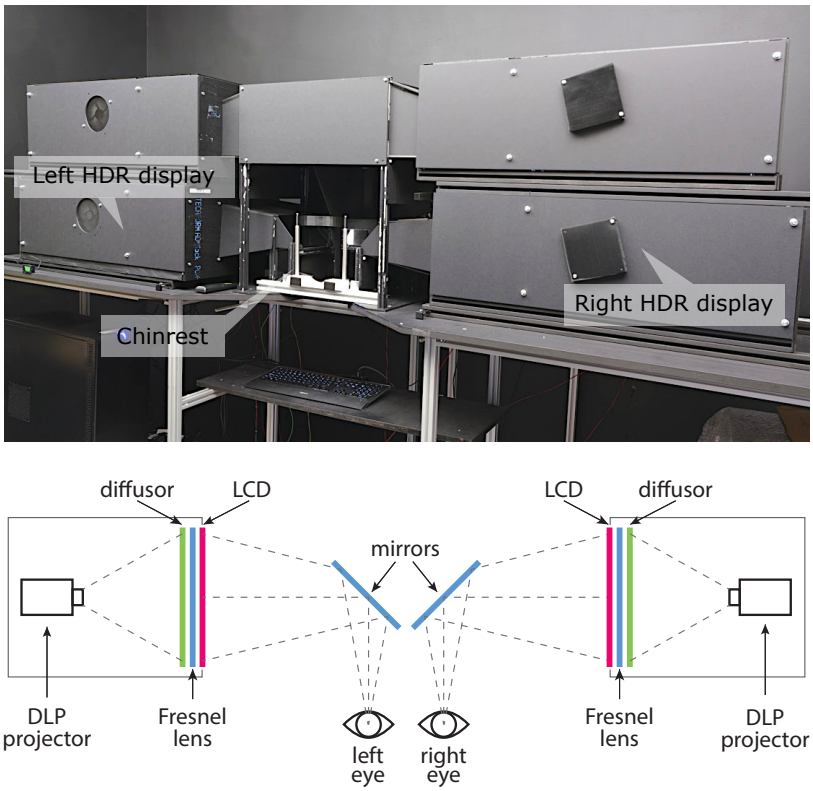


FIGURE 7.2: *Top: The photograph of a prototype stereo HDR display. Each observer sat at the display with their head stabilized by both the chin-rest and the head-rest. Note that the upper part of the display, intended for multi-focal-plane presentation, was not used in this project. Bottom: The schematic diagram of the display, showing the portion of the stereo display.*

Stimuli To study the influence of low luminance on binocular depth perception, we designed a stimulus that contained a controlled binocular disparity cue while minimizing the effect of other depth cues. The observers were presented with a concave, hinge-like shape on a uniform gray background. The stimuli were textured with a procedurally generated pattern (see Figure 7.1, right) with only 2 shades of gray. To isolate only the disparity cue, the texture was projected on a surface from a position of a cyclopean eye, thus eliminating the perspective projection cue (the texture density did not change with the distance). It was also rendered without a reflection model to remove shading cues. During the experiment, observers were asked to use a chinrest to prevent head movements. The stimuli and setup are presented in Figure 7.1.

It is well known that contrast sensitivity strongly changes in the scotopic and mesopic range, hence we decided to investigate the effect for luminances of 0.1, 1 and 10 $\mathrm{cd/m^2}$. However, recent work by Wuerger et al. [2020] has shown that there are also significant changes in contrast sensitivity of above 100 $\mathrm{cd/m^2}$ which motivated is to test two additional luminance levels of 100 and 1000 $\mathrm{cd/m^2}$. For each luminance, 4 contrast levels of the texture were measured: 0.05, 0.1, 0.2, or 0.4. The Weber contrast of the texture is expressed as:

$$C_w = \frac{Y_{max} - Y_{min}}{Y_{min}}, \tag{7.1}$$

where Y_{max} and Y_{min} are maximum and minimum luminance of the stimuli. In the case of our stimuli, Y_{min} was the luminance of the background.

Because in the pilot experiment observers were not able to see the stimuli at $0.1\,\mathrm{cd/m^2}$ and Weber contrast of 0.05, we decided to remove this condition from the main experiment. The order of conditions was randomized for each observer. If the luminance decreased between two conditions, we displayed a uniform field with the target luminance for a minute to ensure the observer was adapted to the new luminance level.

Experimental Procedure The task was to assess whether the angle is greater or smaller than 90 degrees and confirm the decision by pressing a corresponding key on the keyboard. For each of the 19 conditions, we used the method of constant stimuli to estimate the probability of judging the angle as acute or obtuse. The tested angles were: 65°, 75°, 85°, 95°, 105°, and 115°. Six trials were collected for each angle and each observer. We changed the angle by moving the hinge part towards and away from the observer while keeping the side edges stationary (see Figure 7.1). This was done to avoid additional depth cues. The tested angles were randomized between the trials.

Each observer was asked to complete a training session at $10\,\mathrm{cd/m^2}$, in which they were given feedback on whether their answer was correct. Such feedback was not given in the main experiment. The training session helped the observers to familiarize themselves with the task, get accustomed to disparity-only stimuli and was also used to screen observers. We excluded three observers who were unable to do the task in the training session from further experiments. The entire experiment took each participant around two hours and was split into 3–5 short sessions.

Observers Eleven volunteers, who passed the training session, (four females and seven males, aged 24 to 36, including three authors) participated in the experiment. All were recruited from among students and researchers working in the field of computer graphics. Nine of them completed trials for all luminance levels, while two completed only the trials for luminance levels from $0.1\,\mathrm{cd/m^2}$ to $10\,\mathrm{cd/m^2}$. All observers had a normal or corrected-to-normal visual acuity (self-reported). All passed the Titmus stereoacuity test. All observers except the authors were naïve to the purpose of the experiment. Before the experiment, each observer read and signed the consent form. The observers were rewarded for their participation. The experiment was approved by the departmental ethics board.

Results The experiment explained how well the observers could see a geometric angle at several luminance and contrast levels. The data averaged over all observers plotted as the probability that an observer reports an obtuse angle, is shown in Figure 7.3 as red stars.

The first important observation is that the psychometric curves formed by the data points cross 50% probability point at about 90-degree angle regardless of luminance and contrast. This means that the perceived angles were not distorted by lower luminance and contrast. However, the slopes of the psychometric functions differ substantially between the conditions. The shallower slopes indicate that the observers more often mistook the angle at low luminance and low contrast. This means that lower luminance does not reduce the accuracy of the shape assessment task, but it reduces the precision of that task.

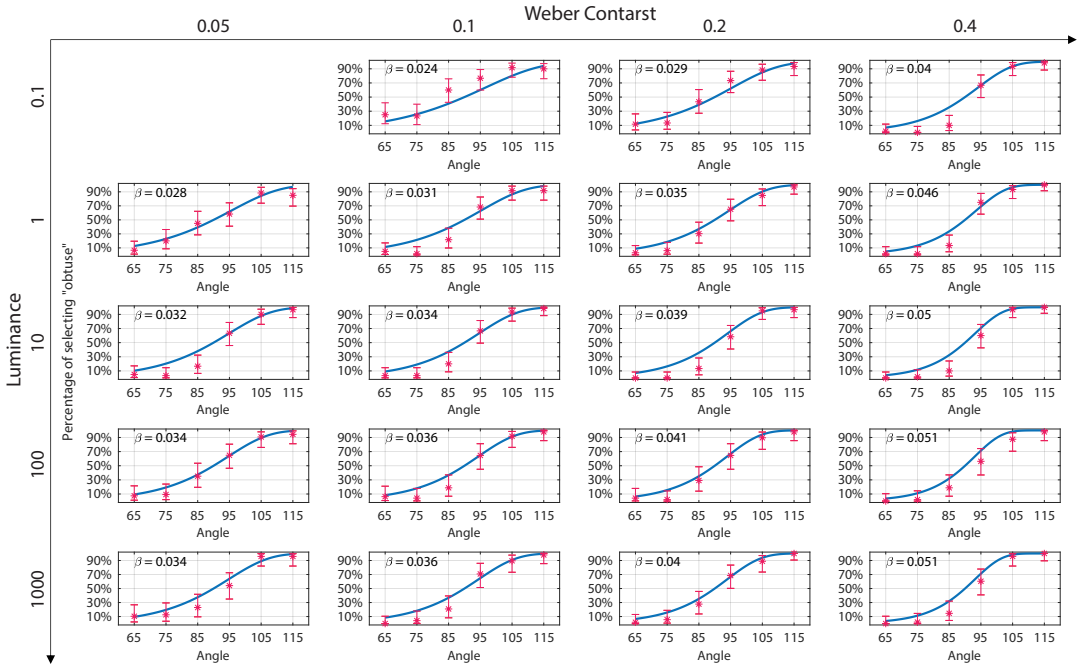


FIGURE 7.3: *The pink stars are the original data points collected from the 3D shape perception experiment (Section 7.2). They represent the frequency at which the participants assessed the angle as obtuse under various luminance and contrast conditions. The error bars denote the 99% confidence intervals. The blue curves represent our fitted psychometric model (Section 7.3). The results for the lowest contrast and luminance are not present as it was impossible to see the stimuli in this condition.*

7.3 Stereo Constancy Model

In this section, we propose a model that can predict how contrast needs to be altered to preserve the same precision of the stereo task across different luminance levels. As the first step, we assume the collected data can be explained by a psychometric function that follows the Weibull cumulative distribution function [Wichmann and Hill, 2001]. We used this psychometric function to describe the probability p of an observer perceiving the hinge-like shape (Figure 7.1) as an obtuse angle given the actual angle α, represented in degrees:

$$p(\alpha, \beta) = 1 - \exp\left(\log(0.5)10^{\beta(\alpha - \alpha_{thr})}\right) \tag{7.2}$$

where α_{thr} is the angle at which the probability of detection p is 0.5, which we assumed to be 90 degrees. The value of β controls the steepness of the function, which reflects the precision of the user performance in this task. A higher value of β means that the observer is more sensitive to the variations in the perceived angle and also that the task is easier.

Our goal is to find a model of β as a function of contrast and luminance, such that the likelihood of the data observed in the 3D shape perception experiment is maximized. We found that β can be explained by a quadratic function of contrast and log-luminance:

$$\beta(c, L; \mathbf{w}) = w_1 L + w_2 c + w_3 L^2 + w_4 c^2 + w_5 \tag{7.3}$$

where L is the logarithm of luminance ($L = \log_{10}(Y)$), c is logarithmic contrast, and $\mathbf{w} = [w_1, ..., w_5]$ denote unknown free parameters. Note that contrast was recorded as Weber contrast C_w Equation 7.1 in our experiment. However, our contrast

enhancement method (Section 7.4) can be implemented more efficiently if it operates on logarithmic contrast. The Weber contrast C_w can be converted into logarithmic contrast c with the formula:

$$c = \log_{10}(C_w + 1). \tag{7.4}$$

To have a better control over free parameters, and to ensure that the function is monotonic, we used the maximum a posteriori (MAP) estimation to find the values of $\mathbf{w}$. We assume $\mathbf{w} \sim N(\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma})^2)$ for some $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, where μ_i and σ_i^2 are the mean and the variance of w_i respectively.

The likelihood of observing k out of n trials (of selecting obtuse angle) can be explained by a binomial distribution, with a latent probability of p of perceiving the angle as obtuse. As indicated by Equation 7.2, the value of p is dependent on the presented angle α and detection sensitivity β, which is then parameterized by contrast c, luminance L, and $\mathbf{w}$ in Equation 7.3. Under the MAP framework, free parameters $\mathbf{w}$ can be found by minimizing the negated log-likelihood of the binomial distribution:

$$\arg\min_{\mathbf{w}} -\sum_s \sum_{\mathbf{d}} \log\left(\binom{n_{s,\mathbf{d}}}{k_{s,\mathbf{d}}} p_{\mathbf{d}}^{k_{s,\mathbf{d}}} \left(1 - p_{\mathbf{d}}^{(n_{s,\mathbf{d}} - k_{s,\mathbf{d}})}\right)\right) + \sum_{i=3,4} \frac{1}{2\sigma_i^2}(w_i - \mu_i)^2 \tag{7.5}$$

where s is the index of the observer, $\mathbf{d} = [\alpha, c, L]$ are the parameters of each condition, and $p_{\mathbf{d}} = p(\alpha, \beta(c, L; \mathbf{w}))$ is given by Equations 7.2 and 7.3. $n_{s,\mathbf{d}}$ in the binomial coefficient is the total number of measurements collected for observer s and condition $\mathbf{d}$ and $k_{s,\mathbf{d}}$ is the number of measurements in which obtuse angle was selected. Table 7.1 shows the final estimated parameters we found for $\mathbf{w}$, and our choices for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. Under the MAP framework, $\frac{1}{2\sigma_i^2}$ becomes the weights on the regularization terms. Note that we only regularize 2^{nd}-order terms w_3 and w_4 to ensure monotonicity. With these parameters, we plot the corresponding fitted psychometric functions parameterized by β under various luminance and contrast conditions on top of the original data points in Figure 7.3 as blue curves. The plots demonstrate that the model explains well most conditions. The worse fit for some conditions (e.g. $0.1\,\mathrm{cd/m^2}$, $C_W = 0.4$) is due to the regularization, which was necessary to make the model monotonic and thus invertible.

Next, we use the fitted model to find the lines of equal precision of the task (constant β). Such lines are plotted as continuous lines in Figure 7.4. The figure shows that to maintain the same precision of the task (the same β), we need to increase the contrast at low luminance and that such an increase should be smaller for higher contrast. We will refer to this model as a stereo constancy model and use it in the next section to derive our contrast enhancement technique for dark stereo displays. To demonstrate that the effect cannot be predicted by a contrast constancy model, we plot in the same figure the contrast constancy model of Kulikowski [Kulikowski, 1976] (used in [Wanat and Mantiuk, 2014]). The comparison shows that stereo constancy requires stronger contrast enhancement between 0.1 and $10\,\mathrm{cd/m^2}$ (mesopic and photopic range) than contrast constancy. The difference between both models will be further corroborated in our validation experiments in Section 7.5.

7.4 Stereo-Preserving Contrast Enhancement Method

We use our model to design a local contrast enhancement method that preserves the precision and difficulty of stereo perception at low luminance. Our method is inspired by the compensation for dark displays proposed by Wanat et al. [2014], yet it differs in its goal as our method is meant to improve stereo perception rather

TABLE 7.1: *Estimated values of free parameters of Equation 7.3 and the priors for the Maximum a Posteriori (MAP) estimation. Symbol "/" means that no prior was used.*

	1	2	3	4	5
w	0.0050	0.1849	-0.0010	0.3994	0.0263
μ	/	/	-0.4	0.4	/
σ^2	/	/	0.001	0.001	/

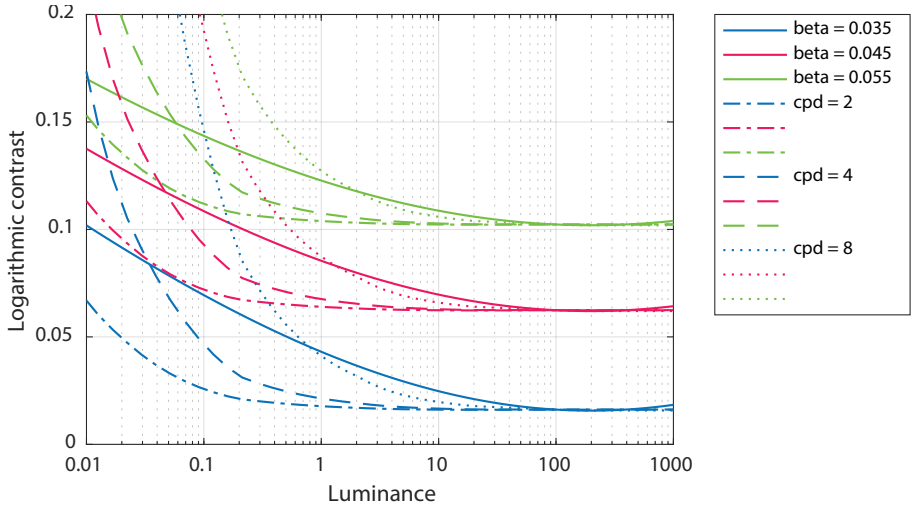


FIGURE 7.4: *The solid lines, or equivalent-β lines, connect the contrast values that result in the same precision of perceiving depth (the same β of the psychophysical function) as a function of different display luminance levels. The lines are derived from our model of stereo task difficulty. The dashed lines represent the equivalent perceived contrast for three different spatial frequencies (2, 4, and 8 cpd) according to Kulikowski's model.*

than contrast and color appearance. We employ the stereo constancy model from the previous section rather than Kulikowski's contrast matching model. We also improve on a few processing steps for better real-time performance and temporal stability. The processing diagram of the method is presented in Figure 7.5. The following subsections present the details of each step of our algorithm.

7.4.1 Color Space Transformation

Because our stereo constancy model is defined in terms of physical (linear) luminance units, we need to convert the rendered frame from a gamma-encoded to a linear color space. Assuming ITU-R BT.709-6 RGB primaries and the standard gamma ($\gamma = 2.2$), the relative luminance, y_{input}, is computed as:

$$y_{\text{input}}(\mathbf{x}) = \sum_{k=1}^{3} v_k \, I'^{\gamma}_{input}(\mathbf{x}, k)\,, \tag{7.6}$$

where $I'(\mathbf{x}, k)$ is the gamma-encoded input value at pixel $\mathbf{x}$ and in color channel k (in the range 0–1), while $v_k = [0.212656, 0.715158, 0.072186]$. Note that we use lower-case y for relative luminance to make it distinct from absolute luminance, Y.

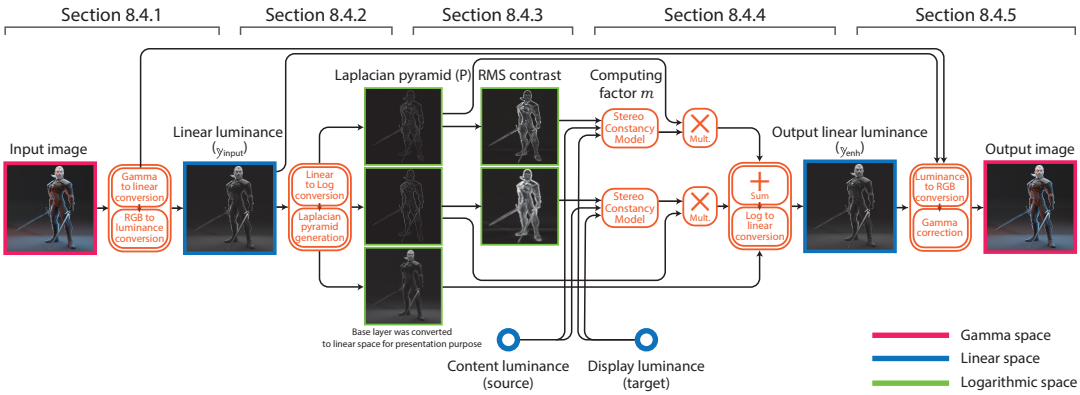


FIGURE 7.5: *Schematic diagram of the proposed method. As the first step, the input image is linearized and color channels are used to compute luminance. Then, a Laplacian pyramid is built using the luminance converted to logarithmic space. Next, we compute a magnitude of contrast which is fed to the stereo constancy model. Per-pixel enhancement factor m produced by the model is multiplied with the contrast acquired in the process of Laplacian pyramid creation. Later all the levels of the pyramid are summed together and converted back to linear space. As the last step, the enhanced luminance channel is combined with the color of the input image, and the image is gamma encoded.*

7.4.2 Multi-Scale Decomposition

The proposed method compensates for the deteriorated depth perception by enhancing local image contrast. In order to operate on local image contrast, we decompose an image into frequency bands using the Laplacian pyramid. We use the classical Burt and Adelson method [Burt and Adelson, 1983] with the coefficient $a = 0.4$ used to construct the filters. In our implementation, we construct a Laplacian pyramid consisting of 3 levels: two band-pass levels and one low-pass level (baseband). The two band-pass levels are sufficient because of the limited effective resolution of VR headsets (in terms of pixels per degree). Such a shallow decomposition also improves the performance in real-time applications.

For computational convenience, the decomposition is performed on logarithmic values of luminance $l = \log_{10}(y_{\text{input}})$. This ensures that the coefficient of the pyramid represents logarithmic contrast (they approximate the logarithm of ratios between two levels). The Laplacian pyramid coefficient at level i is then computed as:

$$P_i(\mathbf{x}) = (g_i * l)(\mathbf{x}) - (g_{i+1} * l)(\mathbf{x}), \tag{7.7}$$

where g_i is the kernel of a Gaussian pyramid at the level i and $*$ is the convolution operator. The pyramid was not subsampled in our implementation. Instead, we used dilated convolutions.

7.4.3 Measure of Local Contrast

Our stereo constancy model requires an estimate of the local contrast to find a corresponding equivalent contrast in the target image. Although we could use the coefficients of the Laplacian pyramid for this purpose, this results in over-enhancement and artifacts at sharp contrast edges, as explained in [Wanat and Mantiuk, 2014] and shown in Figure 7.6. We follow the same approach as in [Wanat and Mantiuk, 2014] and compute a root-mean-squared (RMS) measure of local contrast, but we do it more efficiently by reusing the Gaussian pyramid from the multi-scale decomposition step.

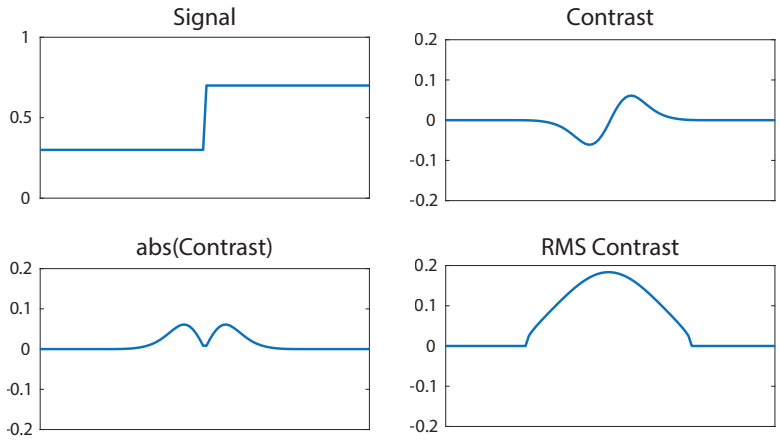


FIGURE 7.6: *Comparison of two different local contrast measures. When a sharp-contrast edge (top-left) is decomposed on a Laplacian pyramid, the band-pass levels will contain a signal that resembles a derivative of a Gaussian (top-right). Using the absolute values on that signal (bottom-left) as a measure of contrast has two disadvantages: (a) the measure has zero-crossing at the edge location; and (b) it underestimates the contrast of the edge as it considers the signal in a single band. Using RMS contrast (bottom-right) solves both problems.*

The localized root-mean-square (RMS) contrast can be computed as:

$$c_i(\mathbf{x}) = \sqrt{(g_\sigma * l^2)(\mathbf{x}) - ((g_\sigma * l)(\mathbf{x}))^2}\,, \tag{7.8}$$

where l is the logarithm of relative luminance and g is a Gaussian kernel with standard deviation σ. We want to use the kernels with larger σ at the lower frequency pyramid levels. To avoid computing additional convolutions, we can instead reuse the Gaussian pyramid and estimate the local RMS contrast as:

$$c_i(\mathbf{x}) = \sqrt{H_i(\mathbf{x}) - G_i^2(\mathbf{x})}\,, \tag{7.9}$$

where G_i is a Gaussian pyramid built from log-luminance l and H_i is a Gaussian pyramids built from squared log-luminance l^2. Computing a second pyramid H is inexpensive on a GPU, as it can be done by operating on a 2-channel texture, where the first channel contains log-luminance and the second channel contains squared log-luminance.

7.4.4 Contrast Retargeting

Once the Laplacian pyramid and contrast magnitude are computed, we can map the contrast for a given source luminance (Y_{in}) to the contrast that provides the same stereoacuity when seen at target luminance (Y_{out}). This can be done by executing the following steps for every frequency band except the low-pass band (baseband), which does not encode contrast.

Finding Contrast Enhancement Factor We need to find an equivalent contrast at another (target) luminance level, which results in the same stereo precision (β) as the original contrast. We can rearrange Equation 7.3 to compute the equivalent contrast

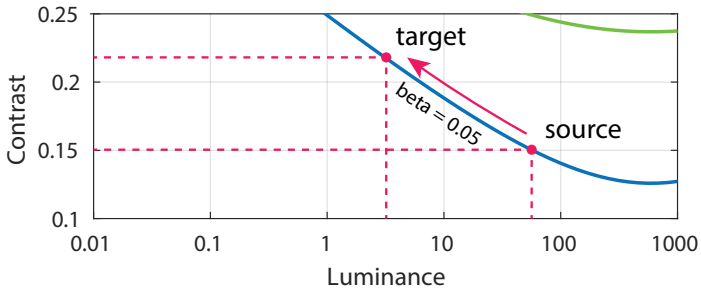


FIGURE 7.7: *Method of finding equivalent contrast that preserves the precision of binocular disparity cues. Similar as in Figure 7.4, for a given input contrast and source luminance, our stereo constancy model gives the curves of equivalent contrast (constant β, blue line). This lets us find the desired contrast for any target display luminance.*

c_{eq} for the desired logarithmic contrast c, source (Y_{in}) and target (Y_{out}) luminance:

$$c_{eq}(c, Y_{\text{in}}, Y_{\text{out}}) = \frac{-w_2 + \sqrt{w_2^2 - 4w_4 t}}{2w_4}, \tag{7.10}$$

where

$$\begin{aligned} t &= w_1 L_{\text{out}} + w_3 L_{\text{out}}^2 + w_5 - \beta(c, L_{\text{in}}) \\ L_{\text{in}} &= \log_{10} Y_{\text{in}} \qquad L_{\text{out}} = \log_{10} Y_{\text{out}} \end{aligned} \tag{7.11}$$

with the parameters $w_1, ..., w_5$ reported in Table 7.1. Function $\beta(\cdot)$ is given in Equation 7.3. The process of mapping contrast between luminance levels is further illustrated in Figure 7.7.

Instead of directly modifying contrast in the Laplacian pyramid, we compute a contrast enhancement factor:

$$m_i(\mathbf{x}) = \frac{c_{\text{eq}}\Big(c_i(\mathbf{x}), Y_{\text{in}}(\mathbf{x}), Y_{\text{out}}(\mathbf{x})\Big)}{c_i(\mathbf{x})}, \tag{7.12}$$

where c_i is an input RMS contrast computed according to Equation 7.9 and $c_{\text{eq}}()$ is the equivalent contrast function from Equation 7.10. Y_{in} and Y_{out} are source and target luminance which are computed as:

$$\begin{aligned} Y_{\text{in}}(\mathbf{x}) &= 10^{G_N(\mathbf{x})} \cdot Y_{\text{peak,src}}\,, \\ Y_{\text{out}}(\mathbf{x}) &= 10^{G_N(\mathbf{x})} \cdot Y_{\text{peak,trg}}\,, \end{aligned} \tag{7.13}$$

where G_N is the base-band of the Gaussian pyramid (as explained in Section 7.4.2). $Y_{\text{peak,src}}$ is the peak luminance of the source display (before dimming) and $Y_{\text{peak,trg}}$ is the peak luminance of the target (dimmed) display. We use $Y_{\text{peak,src}} = 80\,\text{cd/m}^2$ in all our experiments.

Contrast Enhancement Given the local contrast estimate computed in Section 7.4.2, we retarget it, enhancing locally the Laplacian pyramid:

$$\tilde{P}_i(\mathbf{x}) = P_i(\mathbf{x}) \cdot m_i(\mathbf{x}), \tag{7.14}$$

where P_i is the i-th level of Laplacian pyramid ($i = 1, ..., N-1$, excluding the base-band) and m_i is a corresponding enhancement factor from Equation 7.12.

We reconstruct the resulting enhanced luminance channel y_{enh} by summing all N levels of pyramid $\tilde{P}$ including the base band:

$$y_{\text{enh}}(\mathbf{x}) = 10^{\sum_{i=1}^{N} \tilde{P}_i(\mathbf{x})}. \tag{7.15}$$

7.4.5 Reconstructing Color Image

The enhanced color image I_{enh} is produced by multiplying input color (RGB) image in linear space I_{input} by the ratio of enhanced and input luminances:

$$I_{\text{enh}}(\mathbf{x}, k) = I_{\text{input}}(\mathbf{x}, k) \frac{y_{\text{enh}}(\mathbf{x})}{y_{\text{input}}(\mathbf{x})}, \tag{7.16}$$

where k is the index of the color channel ($k \in \{1, 2, 3\}$). Such an approach may, however, result in out-of-gamut colors (one of the color channels values greater than 1) and distorted or desaturated colors. To prevent this, we compute how much a particular pixel can be enhanced until the pixel exceeds the gamut:

$$m_{\max}(\mathbf{x}) = \frac{1}{\max_k \left\{ I_{input}(\mathbf{x}, k) \right\}}. \tag{7.17}$$

Next, we introduce this term into the previous gray to color conversion equation:

$$I_{\text{enh}}(\mathbf{x}, k) = I_{\text{input}}(\mathbf{x}, k) \cdot \min \left\{ \frac{y_{\text{enh}}(\mathbf{x})}{y_{\text{input}}(\mathbf{x})}, m_{\max}(\mathbf{x}) \right\}. \tag{7.18}$$

As the last step we convert the linear color channels to display-ready gamma-encoded ones with a gamma function: $I'(\mathbf{x}, k) = I^{1/\gamma}(\mathbf{x}, k)$. Figure 7.8 shows a comparison between naïve and saturation-aware methods. The left image shows visible, almost white, desaturated petals, while the saturation-aware method preserves colors.

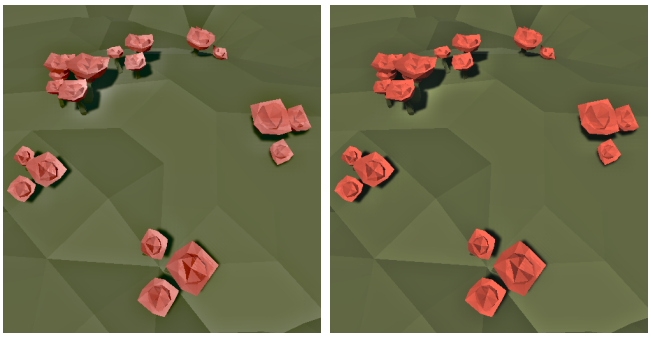


FIGURE 7.8: *Comparison between two methods of grayscale to color conversion. Left: naïve. Right: saturation-aware. Contrast enhancement is exaggerated for low-frequency bands for presentation purposes. The assets are a part of the POLYGON series prepared by Synty Store.*

7.4.6 Implementation Details

We implemented our enhancement method (Section 7.4) in the Unity game engine (v2019.2.19) as a post-processing shader, which took an anti-aliased rendered frame as an input. The code was optimized to ensure that we could run at a stable frame

rate of 90 frames per second on a laptop with an NVIDIA GeForce 1080 graphics card and Intel® Core™ i9-8950HK 2.90 GHz CPU. The performance was measured with the use of Unity built-in profiler.

7.5 Experiment 2: Validation

We evaluated the effectiveness of our method in a validation experiment in which we compared the proposed enhancement algorithm with the most closely related method of Wanat et al. [2014] and standard rendering. The method of Wanat et al. is meant to preserve color and contrast appearance across the luminance range, which can potentially result in an image that also provides better stereoscopic depth cues. The methods were compared in terms of the impression of three-dimensionality and the appearance of the presented scene. We believe that such a preference experiment is required to ensure that proposed method brings practical benefit without any negative effects.

As alternative we considered measuring the response times in a 3D task. However, the design of such an experiment is difficult as the completion time is not only affected by 3D depth perception but also by display and input latency, training, experience, and other effects.

VR Headset The experiments were prepared for the Valve Index VR headset, which offers a relatively high display resolution of a maximum of 16 pixels per visual degree. Additionally, the drivers of this device allow the user to dim its display. Using a luminance meter Minolta LS-100 and manipulating brightness settings, we achieved the desired peak luminance of $5\,\mathrm{cd/m^2}$. While we chose Valve Index due to the convenient settings allowing for display dimming, the effect could be reproduced on other devices (e.g. HTC Vive Pro 2, Oculus Quest 2) after proper dimming (e.g. with the use of neutral density filters).

Stimuli The test scene was built from stylized assets that provided a good balance between good quality content, similar to those found in most VR experiences, and performance (no complex geometry). An example screenshot from the scene is shown in Figure 7.9. Figure 7.10 shows three rendering modes used in the experiment: proposed stereo-constancy model (top), standard rendering (middle), and Wanat's method (bottom). It should be noted that we implemented only the local processing part of the Wanat et al. method to evaluate only that aspect and avoid confounding factors such as changes in color or image brightness. The VR scenes were rendered at 90 Hz.

Procedure During the experiment, we placed the observers in the virtual environment and teleported them to 5 different locations. The initial observers' orientation was randomized, but then they were allowed to look around freely. Additionally, observers were able to switch between two rendering methods using the trackpad on the right controller. In each trial, they compared our method with either standard rendering (no post-processing) or the method of Wanat et al. [2014].

The experiment was split into two parts. In each part, the observers visited the same locations (in random order), regardless of the question being asked. In the first part, the observers were asked to select the rendering mode that looks *more three dimensional* and in the second part they were asked to select the rendering mode that *looks better*. The observers gave the answer by pressing the right trigger while

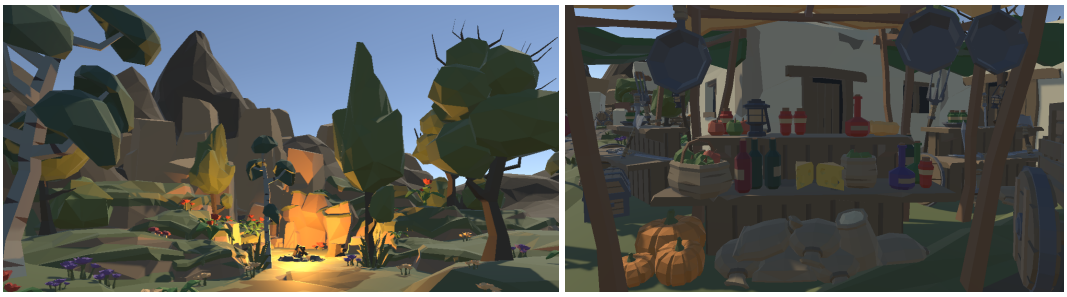


FIGURE 7.9: *Preview of the scene presented to the observer in the preference experiment. The images show a non-enhanced (standard) rendering of the scene. The assets are a part of the POLYGON series prepared by Synty Store.*

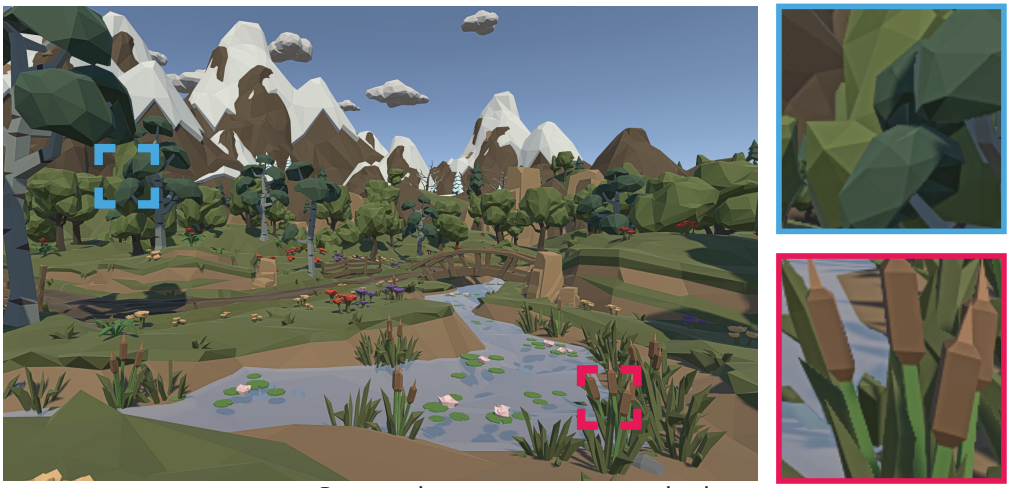
the selected rendering method was active. The participants were allowed to make a selection only after viewing both rendering modes. For both questions and every condition, each location was shown to the participant 5 times, each time using a different direction of the camera (random rotation around the up vector). The order of trials and parts was randomized. We also displayed information about the current progress of the experiment and the assessment criterion (depth or preference) at the bottom of the viewport.

The experiment lasted approximately 20 minutes. According to the post-experiment interviews, it was indicated that the session length was acceptable and did not cause excessive fatigue.

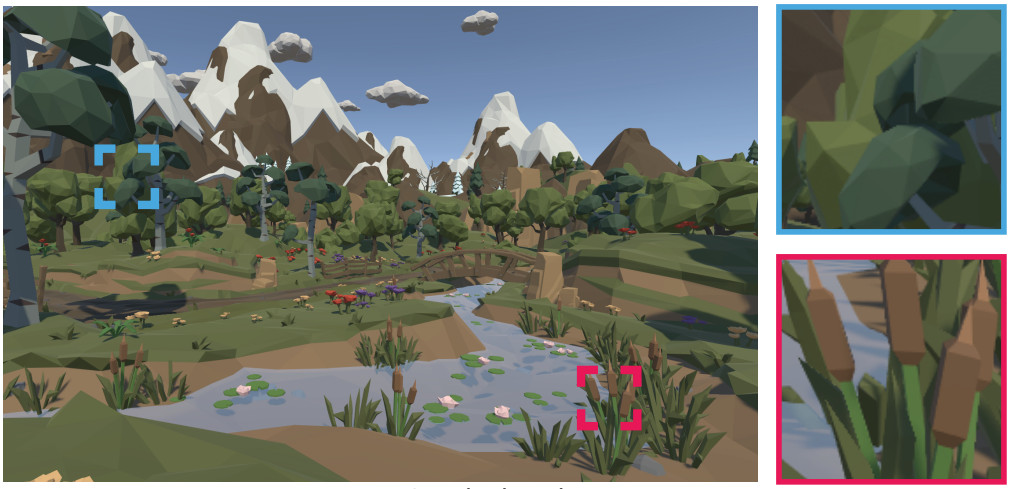
Observers Nine observers (age 22 to 30, 1 female and 8 males) were recruited from among students and researchers working in computer graphics and computer vision. All observers had a normal or corrected-to-normal vision and were also naïve to the purpose of the experiment. Before the experiment, each participant read and signed the consent form. The participants were screened for stereoacuity in a test performed in VR, in which they had to choose a closer square from a pair (akin the Titmus fly test). The results of the observers who failed that test were removed.

Results The bars in Figure 7.11 show the percentage of trials in which our method was chosen over the alternative method. The yellow circles indicate per-observer results. Eight out of nine observers agreed that the image modified with our method looks more three-dimensional and also better than the image enhanced with Wanat et al.'s method and standard rendering. We further validated these results with a one-sided binomial test and a null hypothesis of random selection for both preference and impression of three-dimensionality. The tests confirmed that the results were significant and our contrast enhancement for stereo-constancy improves the perception of 3D shapes and produces more preferred images.

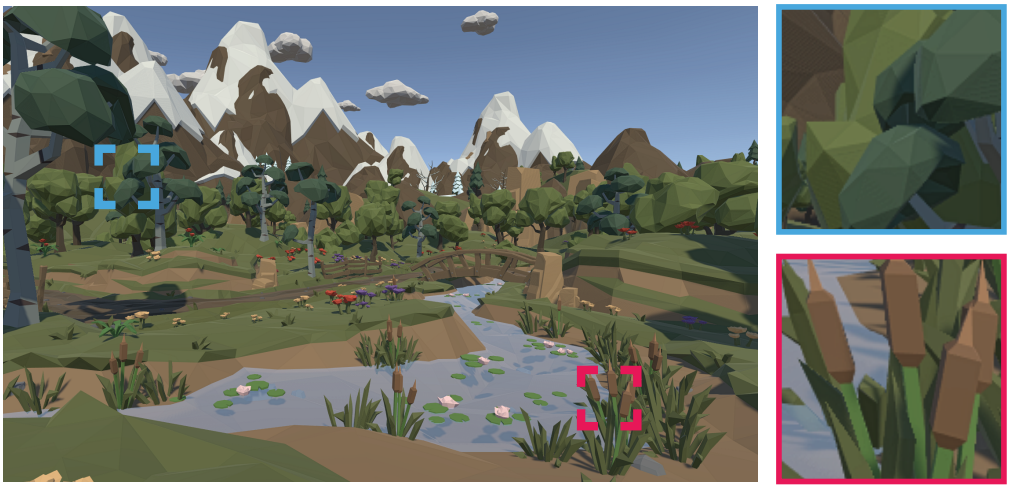
Even though Wanat et al.'s method was intended to improve image appearance, it was not selected when we asked about the preference. The most likely reason for its worse performance is the weaker strength of its enhancement. This is because the strength of the enhancement of that method depends on the changes in the contrast detection threshold (contrast sensitivity function) with luminance. Detection thresholds get smaller very quickly with luminance for high spatial frequencies, but they drop more slowly for low frequencies [Wuerger et al., 2020] (see also Figure 7.4). Since VR headset can reproduce relatively low spatial frequencies (up to 8 cycles per degree), the resulting enhancement was moderate.



Proposed stereo constancy method



Standard rendering



Wanat's method

FIGURE 7.10: *Three rendering methods used in the preference experiment: image enhanced with the proposed stereo-constancy model (top), standard rendering with no enhancement (middle), and image enhanced with Wanat's method (bottom). The insets show close-ups of the selected image areas. It can be observed that Wanat's and the proposed stereo constancy methods increase local contrast and result in a sharper image. The assets are a part of the POLYGON series prepared by Synty Store.*

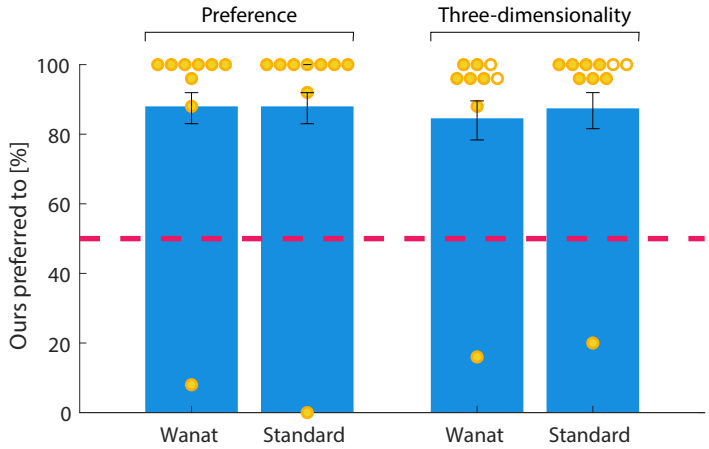


FIGURE 7.11: *Results of Experiment 2, assessing the preference and the impression of three-dimensionality. The yellow circles represent the per-observer results and the empty circles denote observers who failed the stereoacuity test. The height of the bars presents the percentage of trials in which our method was chosen over the method given below (excluding disqualified observers). Standard stands for a no enhancement and Wanat for the method proposed by Wanat and Mantiuk [2014]. The error bars present a 95% confidence interval and the red dashed horizontal line indicates the guess rate.*

From the post-experimental verbal survey, we found that observers based their decision on the fact that the preferred rendering mode appeared sharper and much clearer. Additionally, two observers stated that for the rendering with high contrast, colors seemed to be more vivid and one person mentioned that the sharper image looked like it had a higher resolution. Moreover, no unnatural experiences of our method (including frame rate drop or lack of temporal stability) were reported by the observers.

In the survey, we also asked whether the color seen in the VR headset appeared natural. None of the participants reported any problems with color appearance.

7.6 Discussion

No Distortion of Depth The most important conclusion from our 3D shape perception experiment (Section 7.2) is that low luminance levels (0.1–10 cd/m^2) do not distort depth. This is in contrast to the observations of Kellnhofer et al. [2014], who reported compression of depth at low luminance. Our experiment showed that, even at 0.1 cd/m^2, observers could correctly assess the angle without bias (high accuracy), however with larger variance in their responses (lower precision). Had the perceived depth been compressed at low luminance, the results would have been biased towards obtuse angles, as the observers compensated for the reduced disparity. However, we did not experiment with light levels below 0.1 cd/m^2, so we cannot confirm whether the perception of 3D shapes is affected at these luminance levels.

Contrast vs. Disparity Manipulation Several works [Didyk et al., 2011; Didyk et al., 2012b; Kellnhofer et al., 2016] manipulate disparity to improve depth perception. While such an approach is practical for 3D cinema content, it is unsuitable for VR environments, in which depth must be faithfully reproduced to give accurate visual feedback to egomotion. Disparity manipulation in VR is likely to result in conflicting visual and vestibular sensations leading to VR-sickness [Jacobs et al., 2019].

Color Appearance in Mesopic Vision Degradation in stereoacuity is not the only issue of showing VR content at low brightness. It is well-established that color appearance also degrades as luminance decreases to mesopic vision [Barbur and Stockman, 2010; Fu et al., 2012]. Those models could be incorporated into our enhancement technique, as it was done in [Wanat and Mantiuk, 2014], however, we did not find the changes in color to be substantial enough to require additional processing. None of the experiment participants reported an unnatural color appearance in our post-experiment questionnaire (see Section 7.5). Our observation is also supported by the results of Kwak et al. [2003, Fig. 6, 7] who reported negligible changes in colorfulness and hue, measured using magnitude estimation, when the reference white was reduced to only $1\,\mathrm{cd/m^2}$. Larger changes in color appearance can be observed when two luminance conditions are presented to the observer simultaneously in an asymmetric (haploscopic) matching experiment [Shin et al., 2004]. Such artificial presentation, however, is not representative for viewing content on a dimmed VR headset.

Limitations Our model was fitted to the data collected in the luminance range between $0.1\,\mathrm{cd/m^2}$ and $1\,000\,\mathrm{cd/m^2}$, which may limit the ability of our model to generalize to very low luminance levels. We cannot generalize our model to the scotopic levels much below $0.1\,\mathrm{cd/m^2}$, but we argue that such low luminance is less relevant for displays. We also do not consider the influence of tone mapping on depth perception. Since tone mapping often involves contrast compression, we expect increased difficulty of inferring depth from tone-mapped stereo images.

7.7 Conclusions

Dimming a display can be beneficial for VR experience as it reduces the visibility of flicker, saves power, prolongs battery life, and reduces the cost of the device. The major downside of this approach is the reduced sensitivity to stereoscopic depth cues. Contrary to previous works [Kellnhofer et al., 2014], we do not find the distortion of 3D depth at low luminance (0.1-$1\,\mathrm{cd/m^2}$), but instead, we find increased difficulty and lower precision (larger variance) of assessing 3D shapes based on binocular cues. This motivates our method for enhancing contrast at low luminance levels, intended at improving the reliability of stereoscopic depth cues. We demonstrate that such contrast enhancement can be implemented in the real-time rendering of VR environments. We further show the effectiveness of such depth enhancement in a perceptual experiment asking about qualitative aspects of preference and impression of depth. The experiment demonstrates that depth perception can be effectively restored by contrast enhancement and overall image quality can be improved. The proposed method can improve the user experience for VR headsets that need to operate at low power or those that cannot achieve high refresh rates.

Chapter 8

Conclusion

The thesis reaches its conclusion in this chapter, beginning with a summary and discussion of our contributions in Section 8.1. Subsequently, we delve into potential avenues for future research in Section 8.2. Finally, we provide closing remarks in Section 8.3.

8.1 Discussion

In this dissertation, we demonstrated novel solutions to guide data processing algorithms throughout the whole pipeline of traditional computer graphics, with the use of image, mesh, and motion quality metrics, also considering the intricacies of state-of-the-art displaying technologies. Chapter 3 presented an approach of using a convolutional neural network to design a general purpose visibility metric that provides local information on distortion visibility. The metric enables the automated assessment of rendering artifacts but also finds practical applications in other domains like image processing and computational photography. Further, Chapter 4 provided guidelines on how such metric can be fine-tuned to a specific application in order to increase its accuracy further. This process was designed in such a way as to reduce the time and effort required to collect application-specific training data. In Chapter 5, we took one step back and considered distortions occurring on 3D meshes. The outcome of this part is the standardization of the existing mesh-quality metrics but also extending them to account for various viewing distances, a factor that has never been considered before in the domain of mesh quality metrics. In Chapter 6, we inspected visual quality loss hazards associated with displaying animated content on the screen. We derived the content-adaptive metric that is able to assess the most common spatio-temporal artifacts and enveloped it into a real-time model for finding optimal local shading rate and resolution to provide the best possible user experience. Finally, Chapter 7 investigated the degradation of depth perception in virtual reality experienced in low luminance conditions and provided a perception-based model to restore the original precision of the depth cues. The focus of the work discussed in these chapters lies in modeling perceptual phenomena to attain diverse benefits, such as enhanced performance, quality, and user experience. This underscores the significance of perception as a powerful tool for optimizing various aspects of the computer graphics pipeline.

In the following paragraphs, we explore the contributions and acquired insights for each of the previously mentioned categories of perceptual models.

Image Visibility Metrics

The visibility metric developed in Chapter 3 offers superior accuracy and provides an automated method for assessing local artifact visibility. Our approach utilizes a neural

network-based architecture, which can learn low-level human visual phenomena implicitly, surpassing less advanced metrics like SSIM [Wang et al., 2004]. We attribute this success to the careful design of our training dataset and the statistical model used as a loss function during the training process.

Regarding the former aspect, we constructed a dataset consisting of 557 image pairs with manually marked labels on the visibility of distortions. This dataset surpasses previous efforts in scale, as the next largest dataset contains only 37 marked images [Čadík et al., 2012]. Additionally, our dataset encompassed a broader range of image artifacts regarding both type and magnitude. Additionally, we significantly reduced the time required for subjective data collection thanks to our strategy of using multiple levels of artifact magnitude.

As for the statistical model, it was designed to provide a robust fit to the noisy data. Essentially, it models the observed data while considering the stochastic process of marking distortions in the image and taking into account the probability of mistakes, the probability of attending the distorted region, and the probability of detection.

By leveraging these two components, we not only developed our key contribution—the CNN-based metric mentioned earlier—but also trained the existing methods, significantly improving their accuracy. This is particularly important as certain applications may prioritize metrics with fast execution times at the expense of quality. By utilizing the enhanced versions of state-of-the-art metrics, we strike a balance between accuracy and computational performance.

Furthermore, we demonstrated how our deep-learning-based method could be enhanced to account for varying viewing conditions (similarly to HDR-VDP [Mantiuk et al., 2011c]), such as different viewing distances and peak display luminance levels. To achieve this, we proposed extending the dataset and employing a hybrid metric design that combines white-box modeling of the newly considered parameters with previously-used black-box feature extraction and visibility detection.

In order to improve the applicability, in Chapter 4, detailed guidelines on how our general-purpose metric can be fine-tuned for specific tasks, while minimizing the burden of data collection to the necessary minimum. This contribution is particularly important, as the procedure is conceptually non-trivial but can significantly enhance metric accuracy, making it better suited for various applications.

Finally, we presented notable examples of the utility of our metric in four applications. We demonstrated that our CNN metric facilitates visually lossless compression of JPEG images, can determine the maximum downsampling factor for a single-image super-resolution algorithm to reconstruct visually equivalent images, or seamlessly embed imperceptible watermarks into images. Moreover, using the proposed fine-tuning pipeline, it can identify optimal texture resolutions for 3D assets used in computer games.

In conclusion, the advancements made in these two chapters established a solid foundation for the accurate prediction of visible distortions. Our comprehensive dataset and statistical model allowed us to train a new CNN-based metric, improve the existing metrics, and enabled practical applications that showcased the effectiveness of our approach. By addressing the limitations and exploring future enhancements, we anticipate even greater success in the field of distortion visibility assessment.

Mesh Quality Metrics

The study described in Chapter 5 provided valuable insights into how different geometric distortions affect the perceived quality of meshes, with a specific emphasis

on human face representations. To obtain these findings, we created a unique dataset consisting of 100 high-quality face meshes, ensuring a balanced representation across demographics. We then applied deliberate geometric distortions, including various types of noise, smoothing, and simplification, to this dataset. Through the largest user study on mesh distortions to date, we collected over 84,000 subjective judgments. We took into account important perceptual factors such as distance to ensure a comprehensive analysis. Building upon the results of our study, we conducted a thorough evaluation of the diversity and performance of existing mesh quality metrics.

Throughout our research, we have showcased several promising applications of our perceptual database. We proposed a straightforward data-driven retargeting scheme to standardize these metrics and improve their ability to consider different viewing distances. Additionally, we leveraged the enhanced metrics to suggest applications like perceptually-driven mesh compression and level-of-detail rendering (LOD).

In summary, our work focused on understanding the perceptual impact of distortions on 3D geometry. These findings contribute to a deeper understanding of human perception and facilitate the development of more effective techniques for geometry processing, especially in applications that require optimal tradeoffs between cost and quality.

Metric for spatio-temporal Distortion and Display-Related Effects

In Chapter 6, we have developed a perceptual model for motion quality in the context of real-time graphics. Our goal was to address the challenges posed by the increasing demand for shading computations and the limitations of GPUs, particularly in the face of higher display resolutions and refresh rates. To tackle these challenges, we introduced the Adaptive Local Shading and Refresh Rate (ALSaRR) method, which leverages the Variable Rate Shading (VRS) mechanism and adaptive refresh-rate monitors.

At the core of our ALSaRR method lies the Content-adaptive Metric of Judder, Aliasing, and Blur (CaMoJAB). This metric incorporates psychophysical models of contrast sensitivity, taking into account display characteristics. It has been carefully calibrated using diverse datasets, including our newly created dataset obtained from a subjective quality experiment. CaMoJAB effectively captures the influence of artifacts such as judder, aliasing, and blur, while also considering the mitigating effects of factors like hold-type blur, eye motion blur, and the visual system's inherent limitations in perceiving spatio-temporal information.

We have demonstrated that our ALSaRR method, guided by the CaMoJAB metric, offers fine-grained control over spatio-temporal resolution, maximizing perceived quality within a given rendering budget. By intelligently reallocating the resources to the most vital parts of the spatio-temporal domain, based on factors such as texture content, on-screen velocities, luminance, and display persistence, we achieve an optimal trade-off between visual quality and resource utilization. Our findings are supported by a comprehensive validation study and rigorous statistical analysis. The results indicate that our approach benefits traditional adaptive-sync monitors and holds significant potential for enhancing the visual experience in VR headsets and mobile setups.

In conclusion, our work contributes to the field of perceptual models for motion quality. The ALSaRR method, along with the CaMoJAB metric, offers a practical solution for adaptive control of spatio-temporal resolution, addressing the increasing

demands on GPU resources and improving the visual experience for users. Our research opens up opportunities for further advancements in real-time rendering techniques, benefiting various applications ranging from gaming to virtual reality.

Stereo-Constacy Model

In Chapter 7, our main focus was on investigating the effects of display luminance and contrast on depth perception and stereo vision constancy in the context of VR displays. We explored the benefits and drawbacks of using darker displays for VR headsets, considering factors such as power consumption, motion reproduction, and the reduction of flickering.

One of our key findings is that lowering display luminance does not distort depth perception, contrary to previous works that suggested depth compression at low luminance levels [Kellnhofer et al., 2014]. Through a comprehensive 3D shape perception experiment, we demonstrated that even at extremely low luminance levels (0.1-10 cd/m^2), observers were able to accurately assess angles between slants without bias, albeit with reduced precision. This indicates that the perceived depth remained consistent despite decreased luminance, which has significant implications for maintaining depth fidelity in dimmed VR environments.

To address the challenge of reduced sensitivity to stereoscopic depth cues at low luminance, we proposed a model of stereoscopic constancy that predicts the amount of contrast needed to preserve the precision of binocular disparity depth cues across a wide luminance range. Building upon this model, we developed a multi-scale contrast compensation method that enhances depth perception on dimmed stereoscopic displays. This algorithm, implemented as an efficient post-processing GPU shader, offers real-time applications and can be seamlessly integrated into popular VR engines such as Unity.

Our experimental results demonstrated the effectiveness of our contrast enhancement technique in restoring depth perception and improving overall image quality. Participants of the validation study reported a preference for the enhanced depth perception provided by our method compared to non-processed rendering and existing techniques. This validation underscores the potential of our approach to enhance the user experience in low-brightness VR rendering scenarios, particularly for headsets with limited power capabilities or those unable to achieve high refresh rates.

In conclusion, our study contributes to the understanding of depth perception and stereo vision constancy in VR environments. We have provided insights into the impact of display luminance and contrast on depth judgments based on binocular disparity cues. Furthermore, our proposed contrast enhancement algorithm offers a practical solution to overcome the challenges posed by dimmed displays, enabling improved depth perception and image quality. These findings have implications for various applications, including VR, 3D movies, and other contexts where maintaining accurate depth perception is critical.

8.2 Future Work

Within this section, we explore potential avenues for future research. We categorize our discussions according to the specific metric domains and smaller subtopics that have emerged from our conducted work.

Distortion Visibility Assessment

As HDR displays become increasingly common, it would be valuable to extend our metric to encompass the effects encountered when using displays with extremely low or high luminance levels (emphasizing the latter). For instance, glare from very bright areas could potentially mask distortions in neighboring pixels. By incorporating such effects into our metric, we could provide more accurate predictions and better account for the challenges posed by different luminance ranges, thereby improving the overall quality assessment for HDR content.

Building upon the idea behind the FovVideoVDP metric [Mantiuk et al., 2021b], further extensions for varying viewing conditions could be made by incorporating additional input parameters such as eccentricity. By considering the viewer's gaze location, the metric could account for the non-uniform distribution of visual acuity across the visual field. Such extension could enhance the metric's ability to accurately assess visibility, especially in scenarios where foveated rendering techniques are employed.

To further refine our visibility metric, which was designed to assess the probability of noticing the distortion, assuming that observer knows where the distortion occurs, future research could explore the integration of saliency models. Saliency is a mechanism that reduces cognitive processing efforts by steering attention to certain parts of an image. Incorporating information about salient regions would allow us to modulate the metric's predictions based on users' attention and thus would result in a context-aware visibility assessment, making the metric less conservative. Recent work by Krajancich et al. [2023] shows promise in such skillful incorporation of higher-level cognitive aspects in graphics.

Reducing the inference time is an important aspect to consider in future research to enhance the practicality and usability of our visibility metric. The metric requires a substantial amount of time (approximately 3-4 seconds on a customer-level GPU) to produce predictions, preventing it from being used as a perceptual loss function. Improving the metric's efficiency could make it a valuable tool for optimization and decision-making in various applications.

Exploring new trends in machine learning represents an exciting avenue for future research in advancing visibility metrics. While recent NN-based solutions have shown promise in quality assessment tasks, they often lack the ability to account for various viewing conditions. A fruitful direction would be to investigate the integration of explicitly modeled HVS phenomena into these architectures. By leveraging advancements in machine learning and incorporating domain-specific knowledge, we could develop advanced visibility metrics that provide better accuracy, flexibility, and adaptability to different viewing conditions. This line of research would significantly enhance the assessment of visual content quality.

Our method holds the potential for generating accurate data on distortion visibility that can be used to train no-reference (NR) visibility metrics without running costly and time-consuming user studies. Creating the NR counterpart would greatly expand the range of applicability, enabling perception-driven adaptive sampling for offline rendering algorithms, among other applications. By leveraging the insights gained from our visibility metric, one can develop data-driven approaches to assess visibility without relying on reference images, thereby broadening the scope of visibility assessment in various rendering scenarios.

Mesh Quality Assessment

In Chapter 5, we considered mesh quality metrics that consider only mesh topology and are agnostic to the type of materials and textures used to describe surface appearance. As a promising future work direction, one could investigate interactions between geometry, materials, and texture. This investigation would involve studying how certain materials or textures can mask or hide visual distortions, potentially making them less noticeable to HVS. By conducting comprehensive experiments and analyses, we could gain valuable insights into the specific characteristics of material and texture masking effects, paving the way for more accurate and targeted metrics.

Furthermore, it could be beneficial to understand the sensitivity changes between meshes representing arbitrary objects and human faces. Future studies could focus on investigating the perceptual differences in detecting and perceiving distortions on these two distinct types of stimuli. By conducting rigorous experiments and analyzing the sensitivity variations, we could gain insights into how the HVS processes and prioritizes different types of visual information. These findings could stimulate the development of metrics that are tailored to specific object categories and facilitate the accurate assessment of visual quality in various applications.

Given the diverse range of artifacts and the limitations of existing metrics in generalizing to all of them, developing a new metric that explicitly incorporates their specific characteristics would be a significant advancement. We believe that our study provided important insights that enable this direction of future work.

Human Face Perception

While the metrics described in Chapter 5 effectively assess the visibility of artifacts applied to mesh representations of human faces, an intriguing avenue for future research would be to explore how these artifacts impact the perception of a person's identity. Understanding the influence of visual artifacts on perceived identity is crucial in various domains, such as computer graphics, virtual reality, and facial recognition. By conducting experiments that examine the effects of different types of artifacts on identity perception, we can gain insights into the specific visual cues that contribute to the recognition and representation of a person's identity. The findings could have implications for developing improved techniques for facial representation, identity preservation, and realistic virtual human rendering.

Motion and Display Quality

The current solution for content-aware rendering proposed in Chapter 6 requires a preprocessing step, which improves the performance of the method, but also adds complexity to its usage. To overcome this limitation, one could aim to develop a real-time approximation of the solution that can operate directly in screen space. However, this task is challenging because incorporating information from previous frames, which may contain artifacts due to lower resolution shading, could lead to cumulative errors and inaccurate metric predictions over time. Once this obstacle is overcome, it would be possible to remove the limitations regarding incorrect predictions for reflective and transparent objects as well as particle effects, thus allowing for a more efficient distribution of shading rates and producing more visually appealing renderings.

While designing the method, we made a deliberate choice to eliminate the requirement for an eye-tracker, as it is not yet commonly found in gaming setups at the consumer level. However, integrating eye-tracking technology into our method would

undoubtedly bring significant advantages, particularly by incorporating foveated rendering techniques. Foveated rendering leverages the fact that the human eye has higher acuity in the central region of vision compared to the peripheries. By leveraging this characteristic, computational resources could be redistributed by reducing shading rates in the periphery and allocating them to the foveal area, where finer details are more noticeable. The inclusion of eye-tracking and foveated rendering would further optimize rendering efficiency and enhance the perceived image quality.

As mentioned in Section 6.6, our method considers the most relevant factors influencing motion quality but does not account for some higher-level HVS effects. The HVS exhibits phenomena like contrast masking, where the perception of one visual stimulus can be affected by the presence of another. Additionally, as mentioned for the visibility metric, saliency might reduce sensitivity to certain artifacts, depending on their relative location to the most salient features of the scene. By incorporating these HVS effects into the designed perceptual model, our rendering method could produce more visually pleasing results and more accurately allocate computational resources.

Furthermore, the proposed metric provides predictions based on the perception of an average observer. Nonetheless, it is crucial to acknowledge that individual preferences and characteristics of the HVS can vary significantly, affecting the perceived visual quality. To make the method more versatile and applicable to a wider range of users, one might strive to emphasize the need for a robust and easy-to-use calibration method. Such calibration would allow for personalized adjustments, accommodating individual differences in visual perception and making it more effective across diverse user populations.

Looking towards the future, an intriguing direction for improvement lies in optimizing not only the shading resolution but also the display refresh rate locally. Currently, the refresh rate is uniformly adjusted for the entire screen. However, if the existing display hardware allowed for local refresh rate settings, different regions of the screen could benefit from tailored refresh rates based on their specific content and motion characteristics. This optimization would result in an even more fine-grained and efficient allocation of computational resources, thereby improving both the rendering quality and the overall user experience. Although hardware limitations currently restrict this possibility, advancements in display technology may make it achievable in the near future.

Depth Experience in VR

One potential avenue for future research is to extend our stereo constancy model's scope (Section 7.3) beyond the luminance range covered in the subjective study (0.1 $\mathrm{cd/m^2}$- 1 000 $\mathrm{cd/m^2}$), which we considered for existing customer-level VR headsets. It would be interesting to explore how depth perception is influenced by luminance values that fall outside this range. This research could contribute to enhancing the accuracy and realism of depth perception in future HDR VR systems.

Another promising direction for future work is to address the combined factors of deteriorated binocular disparity depth cues, reduced perceived contrast, and color saturation to provide an optimal user experience that remains faithful to the originally designed user experience. Currently, our solution solves only the first issue while partially mitigating the latter, and a more holistic approach considering these three factors jointly might be required to design more immersive and visually captivating virtual reality experiences, while still providing the additional benefits of low luminance stereo displays.

8.3 Closing Remarks

The primary objective of this thesis was to establish accurate and standardized metrics for quality assessment at different stages of graphical pipeline. This chapter presented comprehensive insights and illustrative instances, affirming the significance of perceptual models as effective instruments for automated evaluation across diverse computer graphics algorithms. Our findings strongly support the notion that these tools hold immense potential in shaping the path towards delivering the ultimate user experience in interactive virtual environments.

Bibliography - Own Work

Jindal, Akshay, Krzysztof Wolski, Karol Myszkowski, and Rafał K. Mantiuk [Dec. 2021]. "Perceptual Model for Adaptive Local Shading and Refresh Rate". In: *ACM Trans. Graph.* 40.6. ISSN: 0730-0301. DOI: 10.1145/3478513.3480514. URL: https://doi.org/10.1145/3478513.3480514.

Wolski, K., D. Giunchi, S. Kinuwaki, P. Didyk, K. Myszkowski, A. Steed, and R. K. Mantiuk [2019]. "Selecting texture resolution using a task-specific visibility metric". In: *Computer Graphics Forum* 38.7, pp. 685–696. DOI: https://doi.org/10.1111/cgf.13871. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13871. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13871.

Wolski, Krzysztof, Daniele Giunchi, Nanyang Ye, Piotr Didyk, Karol Myszkowski, Radosław Mantiuk, Hans-Peter Seidel, Anthony Steed, and Rafał K. Mantiuk [Nov. 2018]. "Dataset and Metrics for Predicting Local Visible Differences". In: *ACM Trans. Graph.* 37.5. ISSN: 0730-0301. DOI: 10.1145/3196493. URL: https://doi.org/10.1145/3196493.

Wolski, Krzysztof, Laura Trutoiu, Zhao Dong, Zhengyang Shen, Kevin Mackenzie, and Alexandre Chapiro [Nov. 2022a]. "Geo-Metric: A Perceptual Dataset of Distortions on Faces". In: *ACM Trans. Graph.* 41.6. ISSN: 0730-0301. DOI: 10.1145/3550454.3555475. URL: https://doi.org/10.1145/3550454.3555475.

Wolski, Krzysztof, Fangcheng Zhong, Karol Myszkowski, and Rafał K. Mantiuk [July 2022b]. "Dark Stereo: Improving Depth Perception under Low Luminance". In: *ACM Trans. Graph.* 41.4. ISSN: 0730-0301. DOI: 10.1145/3528223.3530136. URL: https://doi.org/10.1145/3528223.3530136.

Ye, Nanyang, Krzysztof Wolski, and Rafal K. Mantiuk [June 2019]. "Predicting Visible Image Differences Under Varying Display Brightness and Viewing Distance". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bibliography

Adelson, Edward H and James R Bergen [1985]. "Spatiotemporal energy models for the perception of motion". In: *Josa a* 2.2, pp. 284–299.

Adhikarla, Vamsi K., Marek Vinkler, Denis Sumin, Rafal K. Mantiuk, Karol Myszkowski, Hans-Peter Seidel, and Piotr Didyk [2017]. "Towards a quality metric for dense light fields". In: *Computer Vision and Pattern Recognition*.

Akenine-Möller, Tomas, Eric Haines, and Naty Hoffman [2019]. *Real-time rendering*. Crc Press.

Alakuijala, Jyrki, Robert Obryk, Ostap Stoliarchuk, Zoltan Szabadka, Lode Vandevenne, and Jan Wassenberg [2017]. "Guetzli: Perceptually guided JPEG encoder". In: *arXiv:1703.04421*.

Alam, M. M., K. P. Vilankar, David J Field, and Damon M Chandler [July 2014]. "Local masking in natural images: A database and analysis". In: *Journal of Vision* 14.8, p. 22. ISSN: 1534-7362. DOI: 10.1167/14.8.22.

AMD [2021]. *AMD FidelityFX*. URL: https://www.amd.com/en/technologies/radeon-software-fidelityfx.

Amirshahi, Seyed Ali, Marius Pedersen, and Stella X Yu [2016]. "Image quality assessment by comparing CNN features between images". In: *Journal of Imaging Science and Technology* 60.6, pp. 60410–1.

Aydin, Tunc O, Rafał K. Mantiuk, and Hans-Peter Seidel [2008a]. "Extending quality metrics to full luminance range images". In: *Proceedings of SPIE* 6806, 68060B–68060B–10. ISSN: 0277786X. DOI: 10.1117/12.765095. URL: http://link.aip.org/link/PSISDG/v6806/i1/p68060B/s1{\&}Agg=doi.

Aydin, Tunç Ozan, Rafał Mantiuk, Karol Myszkowski, and Hans-Peter Seidel [2008b]. "Dynamic range independent image quality assessment". In: *ACM Transactions on Graphics (TOG)* 27.3, p. 69.

Bagdanov, Andrew D., Alberto Del Bimbo, and Iacopo Masi [2011]. "The Florence 2D/3D Hybrid Face Dataset". In: *Proceedings of the 2011 joint ACM workshop on Human gesture and behavior understanding*, pp. 79–80. DOI: 10.1145/2072572.2072597.

Bailey, Reynold, Cindy Grimm, Christopher Davoli, and Richard Abrams [2007]. "The effect of object color on depth ordering". In.

Barbur, J. L. and A. Stockman [2010]. "Photopic, mesopic and scotopic vision and changes in visual performance". In: *Encyclopedia of the Eye* 3, pp. 323–331.

Barten, Peter G. J. [2003a]. "Formula for the contrast sensitivity of the human eye". In: *Image Quality and System Performance*. Ed. by Yoichi Miyake and D. Rene Rasmussen. Vol. 5294. International Society for Optics and Photonics. SPIE, pp. 231–238. DOI: 10.1117/12.537476. URL: https://doi.org/10.1117/12.537476.

Barten, Peter GJ [2003b]. "Formula for the contrast sensitivity of the human eye". In: *Image Quality and System Performance*. Vol. 5294. SPIE, pp. 231–238.

Barten, P.G.J. [1999]. *Contrast sensitivity of the human eye and its effects on image quality*. SPIE Press.

Baylor, Denis A, BJ Nunn, and JL Schnapf [1984]. "The photocurrent, noise and spectral sensitivity of rods of the monkey Macaca fascicularis." In: *The Journal of physiology* 357.1, pp. 575–607.

Bianco, Simone, Luigi Celona, Paolo Napoletano, and Raimondo Schettini [2016]. "On the use of deep learning for blind image quality assessment". In: *arXiv:1602.05531*.

Blanz, Volker and Thomas Vetter [1999]. "A Morphable Model for the Synthesis of 3D Faces". In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '99. USA: ACM Press/Addison-Wesley Publishing Co., 187–194. ISBN: 0201485605. DOI: 10.1145/311535.311556. URL: https://doi.org/10.1145/311535.311556.

Booth, James, Anastasios Roussos, Evangelos Ververas, Epameinondas Antonakos, Stylianos Ploumpis, Yannis Panagakis, and Stefanos Zafeiriou [2018]. "3D Reconstruction of "In-the-Wild" Faces in Images and Videos". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.11, pp. 2638–2652. DOI: 10.1109/TPAMI.2018.2832138.

Borji, Ali and Laurent Itti [2013]. "State-of-the-Art in Visual Attention Modeling". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, pp. 185–207. URL: https://api.semanticscholar.org/CorpusID:641747.

Bosse, S., D. Maniry, K. R. Müller, T. Wiegand, and W. Samek [2018]. "Deep neural networks for no-reference and full-reference image quality assessment". In: *IEEE Transactions on Image Processing* 27.1, pp. 206–219.

Bosse, Sebastian, Dominique Maniry, Klaus-Robert Mueller, Thomas Wiegand, and Wojciech Samek [2016a]. "Full-reference image quality assessment using neural networks". In: *Int. Work. Qual. Multimdia Exp.*

Bosse, Sebastian, Dominique Maniry, Thomas Wiegand, and Wojciech Samek [2016b]. "A deep neural network for image quality assessment". In: *IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 3773–3777.

Bosse, Sebastian, Dominique Maniry, Thomas Wiegand, and Wojciech Samek [2016c]. "Neural network-based full-reference image quality assessment". In: *Proceedings of the Picture Coding Symposium (PCS)*, pp. 1–5.

Botsch, Mario, Mark Pauly, Leif Kobbelt, Pierre Alliez, Bruno Lévy, Stephan Bischoff, and Christian Rossl [2007]. "Geometric Modeling Based on Polygonal Meshes". In: *ACM SIGGRAPH Course Notes*. DOI: 10.1145/1281500.1281640.

Burt, P. and E. Adelson [1983]. "The Laplacian Pyramid as a Compact Image Code". In: *IEEE Transactions on Communications* 31.4, pp. 532–540.

Čadík, Martin, Robert Herzog, Rafał Mantiuk, Radosław Mantiuk, Karol Myszkowski, and Hans-Peter Seidel [2013]. "Learning to predict localized distortions in rendered images". In: *Computer Graphics Forum*. Vol. 32. 7, pp. 401–410.

Čadík, Martin, Robert Herzog, Rafał K. Mantiuk, Karol Myszkowski, and Hans-Peter Seidel [2012]. "New measurements reveal weaknesses of image quality metrics in evaluating graphics artifacts". In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 31.6, p. 147.

Campbell, Fergus W. and John G. Robson [1968]. "Application of Fourier Analysis to the Visibility of Gratings". In: *The Journal of physiology* 197.3, pp. 551–566. DOI: 10.1113/jphysiol.1968.sp008574.

Cao, Chen, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou [2014]. "FaceWarehouse: A 3D Facial Expression Database for Visual Computing". In: *IEEE Transactions on Visualization and Computer Graphics* 20.3, pp. 413–425. DOI: 10.1109/TVCG.2013.249.

Cao, Dingcai, Joel Pokorny, Vivianne C Smith, and Andrew J Zele [Nov. 2008]. "Rod contributions to color perception: linear with rod contrast". In: *Vision Research* 48.26, pp. 2586–92. ISSN: 1878-5646. DOI: 10.1016/j.visres.2008.05.001.

Carrier, Etienne [2018]. *Procedural World Generation of 'Far Cry 5'*. Game Devlopers Conference.

Chandler, Damon M [2013]. "Seven challenges in image quality assessment: Past, present, and future research". In: *ISRN Signal Processing*, Article ID 905685. DOI: doi:10.1155/2013/905685.

Chapiro, Alexandre, Robin Atkins, and Scott Daly [2019a]. "A Luminance-aware Model of Judder Perception". In: *ACM Transactions on Graphics (TOG)* 38.5, pp. 1–10.

Chapiro, Alexandre, Robin Atkins, and Scott Daly [2019b]. "A Luminance-Aware Model of Judder Perception". In: *ACM Transactions on Graphics* 38.5. ISSN: 0730-0301. DOI: 10.1145/3338696. URL: https://doi.org/10.1145/3338696.

Cheon, Manri, Sung-Jun Yoon, Byungyeon Kang, and Junwoo Lee [2021]. "Perceptual image quality assessment with transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 433–442.

Cignoni, P., C. Rocchini, and R. Scopigno [1998]. "Metro: Measuring Error on Simplified Surfaces". In: *Computer Graphics Forum* 17.2, pp. 167–174. DOI: https://doi.org/10.1111/1467-8659.00236. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-8659.00236.

Cohen, Jonathan, Marc Olano, and Dinesh Manocha [1998]. "Appearance-preserving Simplification". In: *SIGGRAPH '98*, pp. 115–122.

Colenbrander, Roelof Roderick [May 2021]. *Adaptive graphics for cloud gaming*. US Patent App. 16/688,369.

Cosker, Darren, Eva Krumhuber, and Adrian Hilton [Nov. 2011]. "A FACS Valid 3D Dynamic Action Unit Database with Applications to 3D Dynamic Morphable Facial Modeling". In: pp. 2296–2303. DOI: 10.1109/ICCV.2011.6126510.

Cudeiro, D., T. Bolkart, C. Laidlaw, A. Ranjan, and M. J. Black [June 2019]. "Capture, Learning, and Synthesis of 3D Speaking Styles". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 10093–10103. DOI: 10.1109/CVPR.2019.01034. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.01034.

Curcio, Christine A, Kenneth R Sloan, Robert E Kalina, and Anita E Hendrickson [1990]. "Human photoreceptor topography". In: *Journal of comparative neurology* 292.4, pp. 497–523.

Cutting, J. and P. Vishton [1995]. "Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth". In: *Perception of Space and Motion (Handbook Of Perception And Cognition)*. Ed. by W. Epstein and S. Rogers. Academic Press, pp. 69–117.

Dacey, Dennis M and Michael R Petersen [1990]. "Dendritic field size and morphology of midget and parasol ganglion cells of the human retina". In: *Proceedings of the National Academy of Sciences* 87.24, pp. 9412–9416.

Dai, Hang, Nick Pears, William Smith, and Christian Duncan [Nov. 2019]. "Statistical Modeling of Craniofacial Shape and Texture". In: *International Journal of Computer Vision* 128.2, pp. 547–571. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01260-7. URL: https://doi.org/10.1007/s11263-019-01260-7.

Daly, Scott [1993]. "The visible differences predictor: An algorithm for the assessment of image fidelity". In: *Digital Images and Human Vision*. MIT Press, pp. 179–206.

Daly, Scott, Ning Xu, James Crenshaw, and Vikrant J Zunjarrao [2015]. "A psychophysical study exploring judder using fundamental signals and complex imagery". In: *SMPTE Motion Imaging Journal* 124.7, pp. 62–70.

Daly, Scott J. [1992a]. "The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity". In: *Human Vision, Visual Processing, and Digital Display III*. Vol. 1666. International Society for Optics and Photonics, pp. 2–15.

Daly, Scott J [1992b]. "Visible differences predictor: An algorithm for the assessment of image fidelity". In: *Human Vision, Visual Processing, and Digital Display III*. Vol. 1666. International Society for Optics and Photonics, pp. 2–16.

David, Herbert Aron [1963]. *The Method of Paired Comparisons*. Vol. 12. London.

Davis, James, Yi-Hsuan Hsieh, and Hung-Chi Lee [2015]. "Humans perceive flicker artifacts at 500 Hz". In: *Scientific reports* 5, p. 7861.

DeAngelis, Gregory C, Izumi Ohzawa, and Ralph D Freeman [1995]. "Receptive-field dynamics in the central visual pathways". In: *Trends in neurosciences* 18.10, pp. 451–458.

Debattista, Kurt, Keith Bugeja, Sandro Spina, Thomas Bashford-Rogers, and Vedad Hulusic [2018]. "Frame rate vs resolution: A subjective evaluation of spatiotemporal perceived quality under varying computational budgets". In: *Computer Graphics Forum*. Vol. 37. 1. Wiley Online Library, pp. 363–374.

Deering, Michael [1995]. "Geometry Compression". In: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '95. New York, NY, USA: Association for Computing Machinery, 13–20. ISBN: 0897917014. DOI: 10.1145/218380.218391. URL: https://doi.org/10.1145/218380.218391.

Deguy, Sebastien, Rogelio Olguin, and Brad Smith [2016]. *Texturing Uncharted 4: a matter of Substance*. Game Devlopers Conference.

Denes, Gyorgy, Akshay Jindal, Aliaksei Mikhailiuk, and Rafał K. Mantiuk [July 2020]. "A perceptual model of motion quality for rendering with adaptive refresh-rate and resolution". In: *ACM Transactions on Graphics* 39.4. MARRR, p. 133. ISSN: 0730-0301. DOI: 10.1145/3386569.3392411. URL: https://dl.acm.org/doi/10.1145/3386569.3392411.

Didyk, Piotr, Tobias Ritschel, Elmar Eisemann, Karol Myszkowski, and Hans-Peter Seidel [2011]. "A Perceptual Model for Disparity". In: *ACM Transactions on Graphics*. DOI: 10.1145/1964921.1964991. URL: https://doi.org/10.1145/1964921.1964991.

Didyk, Piotr, Tobias Ritschel, Elmar Eisemann, Karol Myszkowski, and Hans-Peter Seidel [2012a]. "Apparent stereo: the Cornsweet illusion can enhance perceived depth". In: *Human Vision and Electronic Imaging XVII*. Ed. by Bernice E. Rogowitz, Thrasyvoulos N. Pappas, and Huib de Ridder. Vol. 8291. International Society for Optics and Photonics. SPIE, pp. 180 –191. DOI: 10.1117/12.907612. URL: https://doi.org/10.1117/12.907612.

Didyk, Piotr, Tobias Ritschel, Elmar Eisemann, Karol Myszkowski, Hans-Peter Seidel, and Wojciech Matusik [2012b]. "A Luminance-Contrast-Aware Disparity Model and Applications". In: *ACM Transactions on Graphics* 31.6.

Drobot, Michal [2020]. "Software-based Variable Rate Shading in Call of Duty: Modern Warfare". In: *ACM SIGGRAPH 2020 Courses*.

Epic Games [2018]. *Optimizing UE4 for Fortnite: Battle Royale*. Game Devlopers Conference.

Erickson, A., K. Kim, G. Bruder, and G. F. Welch [2020]. "Effects of Dark Mode Graphics on Visual Acuity and Fatigue with Virtual Reality Head-Mounted Displays". In: *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 434–442.

Feng, Xiang, Wanggen Wan, Richard Yi Da Xu, Stuart Perry, Pengfei Li, and Song Zhu [2018]. "A Novel Spatial Pooling Method for 3D Mesh Quality Assessment Based on Percentile Weighting Strategy". In: *Computers & Graphics* 74, pp. 12–22. DOI: https://doi.org/10.1016/j.cag.2018.04.005.

Feng, Xiao-Fan [2006]. "LCD motion blur analysis, perception, and reduction using synchronized backlight flashing". In: *Human Vision and Electronic Imaging XI*. SPIE Vol. 6057, pp. M1–14.

Ferwerda, James A., Peter Shirley, Sumanta N. Pattanaik, and Donald P. Greenberg [1997a]. "A Model of Visual Masking for Computer Graphics". In: *SIGGRAPH '97*, pp. 143–152.

Ferwerda, James A., Peter Shirley, Sumanta N. Pattanaik, and Donald P. Greenberg [1997b]. "A Model of Visual Masking for Computer Graphics". In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '97. USA: ACM Press/Addison-Wesley Publishing Co., 143–152. ISBN: 0897918967. DOI: 10.1145/258734.258818. URL: https://doi.org/10.1145/258734.258818.

Frisby, John P and John E W Mayhew [1978]. "Contrast Sensitivity Function for Stereopsis". In: *Perception* 7.4. PMID: 704272, pp. 423–429. DOI: 10.1068/p070423. eprint: https://doi.org/10.1068/p070423. URL: https://doi.org/10.1068/p070423.

Fu, Chenyang, Changjun Li, Guihua Cui, M. Ronnier Luo, Robert W. G. Hunt, and Michael R. Pointer [2012]. "An investigation of colour appearance for unrelated colours under photopic and mesopic vision". In: *Color Research & Application* 37.4, 238–254. ISSN: 0361-2317. DOI: 10.1002/col.20691.

Fuchs, Martin, Tongbo Chen, Oliver Wang, Ramesh Raskar, Hans-Peter Seidel, and Hendrik P.A. Lensch [2010]. "Real-time temporal shaping of high-speed video streams". In: *Computers & Graphics* 34 [5], pp. 575–584.

Georgeson, B Y M A and G D Sullivan [1975]. "Contrast constancy: deblurring in human vision by spatial frequency channels". In: *The Journal of Physiology* 252.3, pp. 627–656.

Goettker, Alexander, Kevin J. MacKenzie, and T. Scott Murdison [June 2020]. "Differences between oculomotor and perceptual artifacts for temporally limited head mounted displays". In: *Journal of the Society for Information Display* 28.6, pp. 509–519. ISSN: 1071-0922. DOI: 10.1002/jsid.912. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jsid.912.

Goodale, Melvyn A and A David Milner [1992]. "Separate visual pathways for perception and action". In: *Trends in neurosciences* 15.1, pp. 20–25.

Griffin, W. and M. Olano [2015]. "Evaluating Texture Compression Masking Effects Using Objective Image Quality Assessment Metrics". In: *IEEE Trans. Vis. and Comp. Graph.* 21.8, pp. 970–979.

Guo, Jinjiang, Vincent Vidal, Irene Cheng, Anup Basu, Atilla Baskurt, and Guillaume Lavoué [2016]. "Subjective and Objective Visual Quality Assessment of Textured 3D Meshes". In: *ACM Trans. Appl. Percept.* 14.2, 11:1–11:20. ISSN: 1544-3558.

Guo, Yudong, Juyong Zhang, Jianfei Cai, Boyi Jiang, and Jianmin Zheng [May 2018]. "CNN-Based Real-Time Dense Face Reconstruction with Inverse-Rendered Photo-Realistic Face Images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP, pp. 1–1. DOI: 10.1109/TPAMI.2018.2837742.

Hekstra, Gerben Johan, Leo Jan Velthoven, and Michiel Adriaanszoon Klompenhouwer [Jan. 2008]. *Motion blur decrease in varying duty cycle*. US Patent 7,317,445.

Herzog, Robert, Martin Čadík, Tunç O. Aydin, Kwang In Kim, Karol Myszkowski, and Hans-Peter Seidel [2012]. "NoRM: No-reference image quality metric for realistic image synthesis". In: *Computer Graphics Forum* 31.2, pp. 545–554.

Hoffman, David M, Vasiliy I Karasev, and Martin S Banks [2011]. "Temporal presentation protocols in stereoscopic displays: Flicker visibility, perceived motion, and perceived depth". In: *Journal of the Society for Information Display* 19.3, pp. 271–297.

Hoffman, David M. and Grace Lee [Mar. 2019]. "Temporal Requirements for VR Displays to Create a More Comfortable and Immersive Visual Experience". In: *Information Display* 35.2, pp. 9–39. ISSN: 0362-0972. DOI: 10.1002/msid.1018. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/msid.1018.

Holly E. Rushmeier Bernice E. Rogowitz, Christine Piatko [2000]. "Perceptual issues in substituting texture for geometry". In: *Proc.SPIE*. Vol. 3959, pp. 3959 –3959 –12.

Hoshi, Hideo, Wei-Li Liu, Stephen C Massey, and Stephen L Mills [2009]. "ON inputs to the OFF layer: bipolar cells that break the stratification rules of the retina". In: *Journal of Neuroscience* 29.28, pp. 8875–8883.

Hubel, David H and Torsten N Wiesel [1962]. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". In: *The Journal of physiology* 160.1, p. 106.

Intel [2019]. *Use Variable Rate Shading (VRS) to Improve the User Experience in Real-Time Game Engines | SIGGRAPH 2019 Technical Sessions*. URL: https://www.slideshare.net/IntelSoftware/use-variable-rate-shading-vrs-to-improve-the-user-experience-in-real-time-game-engines.

Itti, L., C. Koch, and E. Niebur [1998]. "A model of saliency-based visual attention for rapid scene analysis". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.11, pp. 1254–1259. DOI: 10.1109/34.730558.

Jacobs, Jochen, Xi Wang, and Marc Alexa [2019]. "Keep It Simple: Depth-Based Dynamic Adjustment of Rendering for Head-Mounted Displays Decreases Visual Comfort". In: *ACM Trans. Appl. Percept.* 16.3.

Johnson, Paul V, Joohwan Kim, David M Hoffman, Andy D Vargas, and Martin S Banks [2014]. "Motion artifacts on 240-Hz OLED stereoscopic 3D displays". In: *Journal of the Society for Information Display* 22.8, pp. 393–403.

Jokinen, Kimmo P and Wen Nivala [2017]. "65-4: novel methods for measuring VR/AR performance factors from OLED/LCD". In: *SID Symposium Digest of Technical Papers*. Vol. 48. 1. Wiley Online Library, pp. 961–964.

Julesz, Bela [1964]. "Binocular Depth Perception without Familiarity Cues: Random-dot stereo images with controlled spatial and temporal properties clarify problems in stereopsis." In: *Science* 145.3630, pp. 356–362.

Kang, Le, Peng Ye, Yi Li, and David Doermann [2014]. "Convolutional neural networks for no-reference image quality assessment". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1733–1740.

Karađuzović-Hadžiabdić, Kanita, Jasminka Hasić Telalović, and Rafał K Mantiuk [2017]. "Assessment of multi-exposure HDR image deghosting methods". In: *Computers & Graphics* 63, pp. 1–17.

Karis, Brian [2013]. *Real Shading in Unreal Engine 4*. Tech. rep. Epic Games. URL: http://blog.selfshadow.com/publications/s2013-shading-course/karis/s2013_pbs_epic_notes_v2.pdf.

Kellerer, Hans, Ulrich Pferschy, and David Pisinger [2004]. "Multidimensional Knapsack Problems". In: *Knapsack Problems*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 235–283. ISBN: 978-3-540-24777-7. DOI: 10.1007/978-3-540-24777-7_9. URL: https://doi.org/10.1007/978-3-540-24777-7_9.

Kellnhofer, Petr, Piotr Didyk, Karol Myszkowski, Mohamed M. Hefeeda, Hans-Peter Seidel, and Wojciech Matusik [2016]. "GazeStereo3D: Seamless Disparity Manipulations". In: *ACM Transactions on Graphics* 35.4. DOI: 10.1145/2897824.2925866. URL: http://dx.doi.org/10.1145/2897824.2925866.

Kellnhofer, Petr, Tobias Ritschel, Peter Vangorp, Karol Myszkowski, and Hans-Peter Seidel [2014]. "Stereo Day-for-Night: Retargeting Disparity for Scotopic Vision". In: *ACM Trans. Appl. Percept.* 11.3.

Kim, J., H. Zeng, D. Ghadiyaram, S. Lee, L. Zhang, and A. C. Bovik [Nov. 2017]. "Deep Convolutional Neural Models for Picture-Quality Prediction: Challenges and Solutions to Data-Driven Image Quality Assessment". In: *IEEE Signal Processing Magazine* 34.6, pp. 130–141. ISSN: 1053-5888. DOI: 10.1109/MSP.2017.2736018.

Kim, Min H, Tim Weyrich, and Jan Kautz [2009]. "Modeling human color perception under extended luminance levels". In: *ACM SIGGRAPH 2009 papers*, pp. 1–9.

Kindlmann, Gordon, Erik Reinhard, and Sarah Creem [2002]. "Face-Based Luminance Matching for Perceptual Colormap Generation". In: *IEEE Visualization, 2002*, pp. 299–306. DOI: 10.1109/VISUAL.2002.1183788.

Kingma, Diederik P and Jimmy Ba [2014]. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kraemer, Manuel [2018]. "Accelerating your vr games with vrworks". In: *NVIDIAs GPU Technology Conference (GTC)*.

Krajancich, Brooke, Petr Kellnhofer, and Gordon Wetzstein [2023]. "Towards Attention-aware Foveated Rendering". In: *ACM Trans. Graph.* 40 [4].

Krauskopf, John and Jason D Forte [2002]. "Influence of chromaticity on vernier and stereo acuity". In: *Journal of Vision* 2.9, pp. 6–6.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton [2012]. "ImageNet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., pp. 1097–1105.

Kulikowski, J.J. [Jan. 1976]. "Effective contrast constancy and linearity of contrast sensation". In: *Vision Research* 16.12, pp. 1419–1431. ISSN: 00426989. DOI: 10.1016/0042-6989(76)90161-9. URL: http://www.sciencedirect.com/science/article/pii/0042698976901619.

Kuroki, Yoshihiko, Tomohiro Nishi, Seiji Kobayashi, Hideki Oyaizu, and Shinichi Yoshimura [2007]. "A psychophysical study of improvements in motion-image quality by using high frame rates". In: *Journal of the Society for Information Display* 15.1, pp. 61–68.

Kwak, Youngshin, Lindsay William MacDonald, and M. Ronnier Luo [2003]. "Mesopic color appearance". In: *Human Vision and Electronic Imaging VIII*. Ed. by Bernice E. Rogowitz and Thrasyvoulos N. Pappas. Vol. 5007. International Society for Optics and Photonics. SPIE, pp. 161 –169. DOI: 10.1117/12.477371. URL: https://doi.org/10.1117/12.477371.

Lang, Manuel, Alexander Hornung, Oliver Wang, Steven Poulakos, Aljoscha Smolic, and Markus Gross [2010]. "Nonlinear Disparity Mapping for Stereoscopic 3D". In: *ACM Transactions on Graphics* 29.4. ISSN: 0730-0301. DOI: 10.1145/1778765.1778812. URL: https://doi.org/10.1145/1778765.1778812.

Larimer, James, Jennifer Gille, and James Wong [2001]. "41.2: Judder-Induced Edge Flicker in Moving Objects". In: *SID Symposium Digest of Technical Papers*. Vol. 32. 1. Wiley Online Library, pp. 1094–1097.

Larkin, Michéal and Carol O'Sullivan [2011]. "Perception of Simplification Artifacts for Animated Characters". In: *ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*, pp. 93–100.

Lavoué, G., M. Langer, A. Peytavie, and P. Poulin [2018]. "A Psychophysical Evaluation of Texture Compression Masking Effects". In: *IEEE Trans. Vis. and Comp. Graph.*, pp. 1–1.

Lavoué, Guillaume [2011]. "A Multiscale Metric for 3D Mesh Visual Quality Assessment". In: *Computer Graphics Forum*. Vol. 30. 5. Wiley Online Library, pp. 1427–1437. DOI: 10.1111/j.1467-8659.2011.02017.x.

Lavoué, Guillaume, Elisa Drelie Gelasca, Florent Dupont, Atilla Baskurt, and Touradj Ebrahimi [2006]. "Perceptually Driven 3D Distance Metrics with Application to Watermarking". In: *Applications of Digital Image Processing XXIX*. Vol. 6312. International Society for Optics and Photonics, p. 63120L. DOI: 10.1117/12.686964.

Lavoué, Guillaume, Mohamed Chaker Larabi, and Libor Váša [2016]. "On the Efficiency of Image Metrics for Evaluating the Visual Quality of 3D Models". In: *IEEE Transactions on Visualization and Computer Graphics* 22.8, pp. 1987–1999. DOI: 10.1109/TVCG.2015.2480079.

Lavoué, Guillaume and Rafał Mantiuk [2015]. "Quality Assessment in Computer Graphics". In: *Visual Signal Quality Assessment: Quality of Experience (QoE)*. Ed. by Chenwei Deng, Lin Ma, Weisi Lin, and King Ngi Ngan. Springer International Publishing, pp. 243–286.

Le Meur, Olivier, Patrick Le Callet, and Dominique Barba [Jan. 2006]. "Predicting visual fixations on video based on low-level visual features". In: *Vision Res.* 47, pp. 2483–2498.

Legge, Gordon E and John M Foley [1980]. "Contrast masking in human vision". In: *Josa* 70.12, pp. 1458–1471.

Lewis, Michael B. [2001]. "The Lady's Not for Turning: Rotation of the Thatcher Illusion". In: *Perception* 30.6, pp. 769–774. DOI: 10.1068/p3174.

Li, Chenglin, Laura Toni, Junni Zou, Hongkai Xiong, and Pascal Frossard [2017]. "Delay-power-rate-distortion optimization of video representations for dynamic adaptive streaming". In: *IEEE Transactions on Circuits and Systems for Video Technology* 28.7, pp. 1648–1664.

Liao, Xingran, Baoliang Chen, Hanwei Zhu, Shiqi Wang, Mingliang Zhou, and Sam Kwong [2022]. "DeepWSD: Projecting degradations in perceptual space to wasserstein distance in deep feature space". In: *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 970–978.

Lin, Weisi and C.-C. Jay Kuo [2011]. "Perceptual visual quality metrics: A survey". In: *J. Visual Communication and Image Representation*, pp. 297–312.

Lindstrom, Peter and Greg Turk [2000]. "Image-driven Simplification". In: *ACM Trans. Graph.* 19.3, pp. 204–241.

Livingstone, Margaret and David Hubel [1988]. "Segregation of form, color, movement, and depth: anatomy, physiology, and perception". In: *Science* 240.4853, pp. 740–749.

Livingstone, Margaret S. and David H. Hubel [1994]. "Stereopsis and positional acuity under dark adaptation". In: *Vision Research* 34.6, pp. 799–802.

Loop, Charles [1987]. "Smooth Subdivision Surfaces Based on Triangles". In: *Master's thesis, University of Utah, Department of Mathematics*. URL: https://www.microsoft.com/en-us/research/publication/smooth-subdivision-surfaces-based-on-triangles/.

Lubin, Jeffrey [1995]. "Vision models for target detection and recognition". In: ed. by Eli Peli. World Scientific. Chap. A Visual Discrimination Model for Imaging System Design and Evaluation, pp. 245–283.

Luebke, David, Martin Reddy, Jonathan D. Cohen, Amitabh Varshney, Benjamin Watson, and Robert Huebner [2003]. *Level of Detail for 3D Graphics*. Morgan Kaufmann. DOI: https://doi.org/10.1016/B978-155860838-2/50002-9.

Luft, Thomas, Carsten Colditz, and Oliver Deussen [2006]. "Image Enhancement by Unsharp Masking the Depth Buffer". In: *ACM Transactions on Graphics* 25.3, 1206–1213. ISSN: 0730-0301. DOI: 10.1145/1141911.1142016. URL: https://doi.org/10.1145/1141911.1142016.

Luo, Ming Ronnier and Changjun Li [2013]. "CIECAM02 and Its Recent Developments". In: *Advanced Color Image Processing and Analysis*. Ed. by Christine Fernandez-Maloigne. New York, NY: Springer New York, pp. 19–58. ISBN: 978-1-4419-6190-7. DOI: 10.1007/978-1-4419-6190-7_2. URL: https://doi.org/10.1007/978-1-4419-6190-7_2.

Mackin, Alex, Katy C Noland, and David R Bull [2016]. "The visibility of motion artifacts and their effect on motion quality". In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 2435–2439.

Mantiuk, R., K. Myszkowski, and H.-P. Seidel [2004a]. "Visible difference predicator for high dynamic range images". In: *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*. Vol. 3, 2763–2769 vol.3. DOI: 10.1109/ICSMC.2004.1400750.

Mantiuk, Rafał, Alexander Efremov, Karol Myszkowski, and Hans-Peter Seidel [2006]. "Backward Compatible High Dynamic Range MPEG Video Compression". In: *ACM SIGGRAPH 2006 Papers*. SIGGRAPH '06. New York, NY, USA: Association for Computing Machinery, 713–723. ISBN: 1595933646. DOI: 10.1145/1179352.1141946. URL: https://doi.org/10.1145/1179352.1141946.

Mantiuk, Rafal, Kil Joong Kim, Allan G. Rempel, and Wolfgang Heidrich [2011b]. "HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions". In: *ACM Transactions on Graphics (Proc. SIGGRAPH)*. Article 40.

Mantiuk, Rafał, Kil Joong Kim, Allan G. Rempel, and Wolfgang Heidrich [July 2011a]. "HDR-VDP-2: A Calibrated Visual Metric for Visibility and Quality Predictions in All Luminance Conditions". In: *ACM Trans. Graph.* 30.4. ISSN: 0730-0301. DOI: 10.1145/2010324.1964935. URL: https://doi.org/10.1145/2010324.1964935.

Mantiuk, Rafał, Grzegorz Krawczyk, Karol Myszkowski, and Hans-Peter Seidel [2004b]. "Perception-motivated High Dynamic Range Video Encoding". In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 23.3, pp. 730–738.

Mantiuk, Rafał, Allan G. Rempel, and Wolfgang Heidrich [2009]. "Display considerations for night and low-illumination viewing". In: *Proc. of Symposium on Applied Perception in Graphics and Visualization - APGV '09*, pp. 53–58. ISBN: 9781605587431. DOI: 10.1145/1620993.1621005. URL: http://portal.acm.org/citation.cfm?doid=1620993.1621005.

Mantiuk, Rafał K., Maliha Ashraf, and Alexandre Chapiro [June 2022]. "StelaCSF: A Unified Model of Contrast Sensitivity as the Function of Spatio-Temporal Frequency, Eccentricity, Luminance and Area". In: *ACM Trans. Graph.* 41.4. ISSN: 0730-0301. DOI: 10.1145/3528223.3530115. URL: https://doi.org/10.1145/3528223.3530115.

Mantiuk, Rafał K., Gyorgy Denes, Alexandre Chapiro, Anton Kaplanyan, Gizem Rufo, Romain Bachy, Trisha Lian, and Anjul Patney [2021a]. "FovVideoVDP : A visible difference predictor for wide field-of-view video". In: *ACM Transaction on Graphics* 40.4, p. 49. DOI: 10.1145/3450626.3459831.

Mantiuk, Rafał K., Gyorgy Denes, Alexandre Chapiro, Anton Kaplanyan, Gizem Rufo, Romain Bachy, Trisha Lian, and Anjul Patney [July 2021b]. "FovVideoVDP: A Visible Difference Predictor for Wide Field-of-View Video". In: *ACM Trans. Graph.* 40.4. ISSN: 0730-0301. DOI: 10.1145/3450626.3459831. URL: https://doi.org/10.1145/3450626.3459831.

Mantiuk, Rafat, Kil Joong Kim, Allan G. Rempel, and Wolfgang Heidrich [2011c]. "HDR-VDP-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions". In: *ACM Transactions on Graphics (Proc. SIGGRAPH)*. Vol. 30, p. 40.

Marat, Sophie, Tien Ho-Phuoc, Lionel Granjon, Nathalie Guyader, Denis Pellerin, and Anne Guérin-Dugué [May 2009]. "Modelling Spatio-Temporal Saliency to Predict Gaze Direction for Short Videos". In: *International Journal of Computer Vision* 82. DOI: 10.1007/s11263-009-0215-3.

Menzel, Nicolas and Michael Guthe [2010]. "Towards Perceptual Simplification of Models with Arbitrary Materials". In: *Computer Graphics Forum* 29.7, pp. 2261–2270.

Mikhailiuk, Aliaksei, Clifford Wilmot, Maria Perez-Ortiz, Dingcheng Yue, and Rafal Mantiuk [Jan. 2020]. "Active Sampling for Pairwise Comparisons via Approximate Message Passing and Information Gain Maximization". In: *2020 IEEE International Conference on Pattern Recognition (ICPR)*.

Mikkola, Maija, Atanas Boev, and Atanas Gotchev [2010]. "Relative Importance of Depth Cues on Portable Autostereoscopic Display". In: *Proceedings of the 3rd Workshop on Mobile Video Delivery*. MoViD '10. New York, NY, USA: Association for Computing Machinery, 63–68. ISBN: 9781450301657. DOI: 10.1145/1878022.1878038. URL: https://doi.org/10.1145/1878022.1878038.

Miller, S., M. Nezamabadi, and S. Daly [2013]. "Perceptual Signal Coding for More Efficient Usage of Bit Codes". In: *SMPTE Motion Imaging Journal* 122.4, pp. 52–59. ISSN: 1545-0279. DOI: 10.5594/j18290. URL: http://journal.smpte.org/cgi/doi/10.5594/j18290.

Murdison, T. Scott, Christopher McIntosh, James Hillis, and Kevin J. MacKenzie [2019]. "3-1: Psychophysical Evaluation of Persistence- and Frequency-Limited Displays for Virtual and Augmented Reality". In: *SID Symposium Digest of Technical Papers* 50.1, pp. 1–4. DOI: https://doi.org/10.1002/sdtp.12840. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sdtp.12840. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/sdtp.12840.

Nader, Georges, Kai Wang, Franck Hétroy-Wheeler, and Florent Dupont [2015]. "Just Noticeable Distortion Profile for Flat-Shaded 3D Mesh Surfaces". In: *IEEE transactions on visualization and computer graphics* 22.11, pp. 2423–2436. DOI: 10.1109/TVCG.2015.2507578.

Navarro, Fernando, Susana Castillo, Francisco J Serón, and Diego Gutierrez [2011]. "Perceptual considerations for motion blur rendering". In: *ACM Transactions on Applied Perception (TAP)* 8.3, pp. 1–15.

Nelson, Ralph [1973]. "A comparison of electrical properties of neurons in Necturus retina." In: *Journal of Neurophysiology* 36.3, pp. 519–535.

Nouri, Anass, Christophe Charrier, and Olivier Lézoray [2016]. "Full-Reference Saliency-Based 3D Mesh Quality Assessment Index". In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 1007–1011. DOI: 10.1109/ICIP.2016.7532509.

Nvidia [2018]. "NVIDIA Turing GPU Architecture". In: URL: https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf.

Oliva, Aude, Antonio Torralba, and Philippe G. Schyns [July 2006]. "Hybrid Images". In: *ACM Trans. Graph.* 25.3, 527–532. ISSN: 0730-0301. DOI: 10.1145/1141911.1141919.

Ortega, A. and K. Ramchandran [1998]. "Rate-distortion methods for image and video compression". In: *IEEE Signal Processing Magazine* 15.6, pp. 23–50. DOI: 10.1109/79.733495.

O'Shea, James P., Martin S. Banks, and Maneesh Agrawala [2008]. "The Assumed Light Direction for Perceiving Shape from Shading". In: *Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization*. APGV '08. Los Angeles, California: Association for Computing Machinery, 135–142. ISBN: 9781595939814. DOI: 10.1145/1394281.1394306.

Oskam, Thomas, Alexander Hornung, Huw Bowles, Kenny Mitchell, and Markus Gross [2011]. "OSCAM - Optimized Stereoscopic Camera Control for Interactive 3D". In: *ACM Transactions on Graphics* 30.6, 1–8. ISSN: 0730-0301. DOI: 10.1145/2070781.2024223. URL: https://doi.org/10.1145/2070781.2024223.

Pan, Yixin, Irene Cheng, and A. Basu [2005]. "Quality metric for approximating subjective evaluation of 3-D objects". In: *IEEE Transactions on Multimedia* 7.2, pp. 269–279.

Panda, S. S., M. S. R. S. Prasad, and G. Jena [2011]. "POCS based super-resolution image reconstruction using an adaptive regularization parameter". In: *CoRR* abs/1112.1484. arXiv: 1112.1484. URL: http://arxiv.org/abs/1112.1484.

Paysan, Pascal, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter [2009]. "A 3D Face Model for Pose and Illumination Invariant Face Recognition". In: *IEEE international conference on advanced video and signal based surveillance*, pp. 296–301. DOI: 10.1109/AVSS.2009.58.

Peli, Eli, Jian Yang, Robert Goldstein, and Adam Reeves [Aug. 1991]. "Effect of luminance on suprathreshold contrast perception". In: *Journal of the Optical Society of America A* 8.8, p. 1352. ISSN: 1084-7529. DOI: 10.1364/JOSAA.8.001352. URL: https://www.osapublishing.org/abstract.cfm?URI=josaa-8-8-1352.

Pelli, Denis G and Peter Bex [2013]. "Measuring contrast sensitivity". In: *Vision research* 90, pp. 10–14.

Perez-Ortiz, Maria and Rafal K. Mantiuk [Dec. 2017]. "A practical guide and software for analysing pairwise comparison experiments". In: *arXiv preprint*. arXiv: 1712.03686. URL: http://arxiv.org/abs/1712.03686.

Perez-Ortiz, Maria, Aliaksei Mikhailiuk, Emin Zerman, Vedad Hulusic, Giuseppe Valenzise, and Rafał K Mantiuk [2019]. "From Pairwise Comparisons and Rating to a Unified Quality Scale". In: *IEEE Transactions on Image Processing* 29, pp. 1139–1151. DOI: 10.1109/TIP.2019.2936103.

Perlin, Ken [July 2002]. "Improving Noise". In: vol. 21. 3. New York, NY, USA: Association for Computing Machinery, 681–682. DOI: 10.1145/566654.566636. URL: https://doi.org/10.1145/566654.566636.

Piórkowski, Rafał, Radosław Mantiuk, and Adam Siekawa [2017]. "Automatic detection of game engine artifacts using full reference image quality metrics". In: *ACM Transactions on Applied Perception (TAP)* 14.3, p. 14.

Poggio, Gian F and William H Talbot [1981]. "Mechanisms of static and dynamic stereopsis in foveal cortex of the rhesus monkey". In: *The Journal of physiology* 315.1, pp. 469–492.

Pohl, Daniel, Timo Bolkart, Stefan Nickels, and Oliver Grau [2015]. "Using astigmatism in wide angle HMDs to improve rendering". In: *2015 IEEE Virtual Reality (VR)*, pp. 263–264. DOI: 10.1109/VR.2015.7223396.

Ponomarenko, Nikolay, Lina Jin, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, and C.-C. Jay Kuo [Jan. 2015a]. "Image database TID2013: Peculiarities, results and perspectives". In: *Signal Processing: Image Communication* 30, pp. 57–77. ISSN: 09235965.

DOI: 10.1016/j.image.2014.10.009. URL: http://linkinghub.elsevier.com/retrieve/pii/S0923596514001490.

Ponomarenko, Nikolay, Lina Jin, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, et al. [2015b]. "Image Database TID2013: Peculiarities, Results and Perspectives". In: *Signal processing: Image communication* 30, pp. 57–77. DOI: 10.1016/j.image.2014.10.009.

Qu, L. and G. W. Meyer [2008]. "Perceptually Guided Polygon Reduction". In: *IEEE Trans. Vis. and Comp. Graph.* 14.5, pp. 1015–1029.

Qualcomm [2021]. *Qualcomm Snapdragon 888 levels up mobile gaming*. URL: https://www.qualcomm.com/news/onq/2020/12/02/qualcomm-snapdragon-888-levels-mobile-gaming.

Quiroga, Rodrigo Quian [2012]. "Concept cells: the building blocks of declarative memory functions". In: *Nature Reviews Neuroscience* 13.8, pp. 587–597.

Ranjan, Anurag, Timo Bolkart, Soubhik Sanyal, and Michael J Black [2018]. "Generating 3D Faces Using Convolutional Mesh Autoencoders". In: *Proceedings of the European Conference on Computer Vision*, pp. 704–720. URL: http://arxiv.org/abs/1807.10267.

Rejhon, Mark [2017]. *Strobe Crosstalk: Blur Reduction Double-Images*. URL: https://blurbusters.com/faq/advanced-strobe-crosstalk-faq/.

Richards, Whitman [1971]. "Anomalous stereoscopic depth perception". In: *JOSA* 61.3, pp. 410–414.

Roberts, JE and AJ Wilkins [2013]. "Flicker can be perceived during saccades at frequencies in excess of 1 kHz". In: *Lighting Research & Technology* 45.1, pp. 124–132.

Roe, Anna W and DY Ts'o [1995]. "Visual topography in primate V2: multiple representation across functional stripes". In: *Journal of Neuroscience* 15.5, pp. 3689–3715.

Rogowitz, Bernice E. and Holly E. Rushmeier [June 2001]. "Are Image Quality Metrics Adequate to Evaluate the Quality of Geometric Objects?" In: *Human Vision and Electronic Imaging VI*. Vol. 4299. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, pp. 340–348. DOI: 10.1117/12.429504.

Roorda, Austin and David R Williams [1999]. "The arrangement of the three cone classes in the living human eye". In: *Nature* 397.6719, pp. 520–522.

Saunders, Jeffrey A. and Benjamin T. Backus [Aug. 2006]. "Perception of Surface Slant From Oriented Textures". In: *Journal of Vision* 6.9, pp. 3–3. DOI: 10.1167/6.9.3.

Schmid, Jan, Y ULUDAG, and J DELIGIANNIS [2019]. "It just works: Raytraced reflections in" Battlefield V"". In: *GPU Technology Conference*.

Schönemann, Peter H [1966]. "A Generalized Solution of the Orthogonal Procrustes Problem". In: *Psychometrika* 31.1, pp. 1–10. DOI: 10.1007/BF02289451.

Schuchhardt, M., S. Jha, R. Ayoub, M. Kishinevsky, and G. Memik [2015]. "Optimizing mobile display brightness by leveraging human visual perception". In: *2015 International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*, pp. 11–20.

Series, BT [2020]. "The present state of ultra-high definition television". In.

Serrano, Ana, Bin Chen, Chao Wang, Michal Piovarči, Hans-Peter Seidel, Piotr Didyk, and Karol Myszkowski [July 2021]. "The Effect of Shape and Illumination on Material Perception: Model and Applications". In: *ACM Trans. Graph.* 40.4. ISSN: 0730-0301. DOI: 10.1145/3450626.3459813. URL: https://doi.org/10.1145/3450626.3459813.

Sheikh, Hamid R., Zhou Wang, Lawrence Cormack, and Alan C Bovik [2005]. "LIVE Image Quality Assessment Database Release 2". In: URL: http://live.ece.utexas.edu/research/quality.

Sheikh, H.R., M.F. Sabir, and A.C. Bovik [2006]. "A statistical evaluation of recent full reference image quality assessment algorithms". In: *IEEE Trans. on Image Processing* 15.11, pp. 3440–3451. ISSN: 1057-7149.

Shin, Jae Chul, Hirohisa Yaguchi, and Satoshi Shioiri [2004]. "Change of Color Appearance in Photopic, Mesopic and Scotopic Vision". In: *Optical Review* 11.4, 265–271. ISSN: 1340-6000. DOI: 10.1007/s10043-004-0265-2.

Sim, Kyohoon, Jiachen Yang, Wen Lu, and Xinbo Gao [2020]. "MaD-DLS: mean and deviation of deep and local similarity for image quality assessment". In: *IEEE Transactions on Multimedia* 23, pp. 4037–4048.

Simmons, David R and Frederick AA Kingdom [1994]. "Contrast thresholds for stereoscopic depth identification with isoluminant and isochromatic stimuli". In: *Vision Research* 34.22, pp. 2971–2982.

Simons, Daniel J and Ronald A Rensink [2005]. "Change Blindness: Past, Present, and Future". In: *Trends in Cognitive Sciences* 9.1, pp. 16–20. DOI: https://doi.org/10.1016/j.tics.2004.11.006.

Singh, Gurjot, Stephen R. Ellis, and J. Edward Swan [2018]. "The Effect of Focal Distance, Age, and Brightness on Near-Field Augmented Reality Depth Matching". In: *IEEE Transactions on Visualization and Computer Graphics* 26.2, pp. 1385–1398. ISSN: 19410506. DOI: 10.1109/TVCG.2018.2869729. eprint: 1712.00088.

Sluyterman, AAS [2006]. "What is needed in LCD panels to achieve CRT-like motion portrayal?" In: *Journal of the Society for Information Display* 14.8, pp. 681–686.

Sorkine, Olga, Daniel Cohen-Or, and Sivan Toledo [2003]. "High-Pass Quantization for Mesh Encoding". In: *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. SGP '03. Aachen, Germany: Eurographics Association, 42–51. ISBN: 1581136870. DOI: 10.2312/SGP/SGP03/042-051.

Stabell, Bjørn and Ulf Stabell [Nov. 1998]. "Chromatic rod–cone interaction during dark adaptation". In: *J. Opt. Soc. Am. A* 15.11, pp. 2809–2815. DOI: 10.1364/JOSAA.15.002809. URL: http://josaa.osa.org/abstract.cfm?URI=josaa-15-11-2809.

Stengel, M., P. Bauszat, M. Eisemann, E. Eisemann, and M. Magnor [2015]. "Temporal Video Filtering and Exposure Control for Perceptual Motion Blur". In: *Visualization and Computer Graphics, IEEE Transactions on* 21.2, pp. 1–11. DOI: 10.1109/TVCG.2014.2377753.

Sun, Jennifer and Pietro Perona [1998]. "Where Is the Sun?" In: *Nature neuroscience* 1.3, pp. 183–184. DOI: 10.1038/630.

Sung, K., A. Pearce, and C. Wang [2002]. "Spatial-Temporal Antialiasing". In: *IEEE Transactions on Visualization and Computer Graphics* 8.2, pp. 144–153.

Tanaka, Keiji [1996]. "Inferotemporal cortex and object vision". In: *Annual review of neuroscience* 19.1, pp. 109–139.

Thompson, Peter [1980]. "Margaret Thatcher: A New Illusion". In: *Perception* 9.4, pp. 483–484. DOI: 10.1068/p090483.

Tian, Dihong and Ghassan AlRegib [2004]. "FQM: A Fast Quality Measure for Efficient Transmission of Textured 3D Models". In: *ACM MULTIMEDIA '04*, pp. 684–691.

Troscianko, Tom, Rachel Montagnon, Jacques Le Clerc, Emmanuelle Malbert, and Pierre-Louis Chanteau [1991]. "The role of colour as a monocular depth cue". In: *Vision research* 31.11, pp. 1923–1929.

Tursun, Okan Tarhan, Elena Arabadzhiyska, Marek Wernikowski, Radosław Mantiuk, Hans-Peter Seidel, Karol Myszkowski, and Piotr Didyk [2019]. "Luminance-Contrast-Aware Foveated Rendering". In: *ACM Transactions on Graphics (Proc. ACM SIGGRAPH)* 38.4.

United States Census Bureau [2021]. *US census*. [Online; accessed 14-December-2021]. URL: https://www.census.gov/quickfacts/fact/table/US/PST045221.

Unity [2021]. *Unity - Manual: Dynamic resolution*. URL: https://docs.unity3d.com/Manual/DynamicResolution.html.

Unreal [2021]. URL: https://docs.unrealengine.com/en-US/RenderingAndGraphics/DynamicResolution/index.html.

Vaidyanathan, Karthik, Marco Salvi, Robert Toth, Tim Foley, Tomas Akenine-Möller, Jim Nilsson, Jacob Munkberg, Jon Hasselgren, Masamichi Sugihara, Petrik Clarberg, et al. [2014]. "Coarse pixel shading". In: *Proceedings of High Performance Graphics*, pp. 9–18.

Vaidyanathan, Karthik, Robert Toth, Marco Salvi, Solomon Boulos, and Aaron E Lefohn [2012]. "Adaptive Image Space Shading for Motion and Defocus Blur." In: *High Performance Graphics*, pp. 13–21.

Van Essen, David C and John HR Maunsell [1983]. "Hierarchical organization and functional streams in the visual cortex". In: *Trends in neurosciences* 6, pp. 370–375.

Vancleef, Kathleen, Jenny CA Read, William Herbert, Nicola Goodship, Maeve Woodhouse, and Ignacio Serrano-Pedraza [2018]. "Two choices good, four choices better: For measuring stereoacuity in children, a four-alternative forced-choice paradigm is more efficient than two". In: *PloS one* 13.7, e0201366.

Vasylevska, Khrystyna, Hyunjin Yoo, Tara Akhavan, and Hannes Kaufmann [Mar. 2019]. "Towards Eye-Friendly VR: How Bright Should It Be?" In: *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, pp. 566–574. ISBN: 978-1-7281-1377-7. DOI: 10.1109/VR.2019.8797752. URL: https://ieeexplore.ieee.org/document/8797752/.

Váša, Libor and Jan Rus [Aug. 2012]. "Dihedral Angle Mesh Error: A Fast Perception Correlated Distortion Measure for Fixed Connectivity Triangle Meshes". In: vol. 31. 5. Chichester, GBR: The Eurographs Association & John Wiley & Sons, Ltd., 1715–1724. DOI: 10.1111/j.1467-8659.2012.03176.x. URL: https://doi.org/10.1111/j.1467-8659.2012.03176.x.

Verbeure, Tom, Gerrit A Slavenburg, Thomas F Fox, Robert Jan Schutten, Luis Mariano Lucas, and Marcel Dominicus Janssens [Sept. 2017]. *System, method, and computer program product for combining low motion blur and variable refresh rate in a display*. US Patent 9,773,460.

Walls, Gordon Lynn [1944]. *The vertebrate eye and its adaptive radiation*.

Walter, Bruce, Sumanta N. Pattanaik, and Donald P. Greenberg [2002]. "Using Perceptual Texture Masking for Efficient Image Synthesis". In: *Computer Graphics Forum* 21.3.

Wanat, Robert and Rafał K. Mantiuk [2014]. "Simulating and compensating changes in appearance between day and night vision". In: *ACM Trans. Graph.* 33, 147:1–147:12.

Wandell, Brian A [1995]. *Foundations of vision*. Sinauer Associates.

Wang, Kai, Fakhri Torkhani, and Annick Montanvert [2012]. "A Fast Roughness-Based Approach to the Assessment of 3D Mesh Visual Quality". In: *Computers & Graphics* 36.7, pp. 808–818. DOI: https://doi.org/10.1016/j.cag.2012.06.004.

Wang, Zhou, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli [2004]. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4, pp. 600–612. DOI: 10.1109/TIP.2003.819861.

Wang, Zhou and Alan C. Bovik [2006]. *Modern image quality assessment*. Morgan & Claypool Publishers.

Wann, John P, Simon Rushton, and Mark Mon-Williams [1995]. "Natural problems for stereoscopic depth perception in virtual environments". In: *Vision research* 35.19, pp. 2731–2736.

Wässle, Heinz, Ulrike Grünert, Jürgen Röhrenbeck, and Brian B Boycott [1990]. "Retinal ganglion cell density and cortical magnification factor in the primate". In: *Vision research* 30.11, pp. 1897–1911.

Watanabe, M and RW Rodieck [1989]. "Parasol and midget ganglion cells of the primate retina". In: *Journal of Comparative Neurology* 289.3, pp. 434–454.

Watson, Andrew B [2013]. "High frame rates and human vision: A view through the window of visibility". In: *SMPTE Motion Imaging Journal* 122.2, pp. 18–32.

Watson, Andrew B and Albert J Ahumada [1985]. "Model of human visual-motion sensing". In: *JOSA A* 2.2, pp. 322–342.

Watson, Andrew B and Albert J Ahumada [2005]. "A standard model for foveal detection of spatial contrast". In: *Journal of vision* 5.9, pp. 6–6.

Watson, Andrew B and Albert J Ahumada [2011]. "Blur clarified: A review and synthesis of blur discrimination". In: *Journal of Vision* 11.5, pp. 10–10.

Watson, Andrew B. and Denis G. Pelli [1983]. "Quest: A Bayesian adaptive psychometric method". In: *Perception & Psychophysics* 33.2, pp. 113–120.

Watt, Simon J., Kurt Akeley, Marc O. Ernst, and Martin S. Banks [Dec. 2005]. "Focus cues affect perceived depth". In: *Journal of Vision* 5.10, pp. 7–7. ISSN: 1534-7362. DOI: 10.1167/5.10.7. eprint: https://arvojournals.org/arvo/content_public/journal/jov/932834/jov-5-10-7.pdf. URL: https://doi.org/10.1167/5.10.7.

Wichmann, Felix A and N Jeremy Hill [2001]. "The psychometric function: I. Fitting, sampling, and goodness of fit". In: *Perception & Psychophysics* 63.8, pp. 1293–1313.

Wilcox, Laurie M, Robert S Allison, John Helliker, Bert Dunk, and Roy C Anthony [2015]. "Evidence that viewers prefer higher frame-rate film". In: *ACM Transactions on Applied Perception (TAP)* 12.4, pp. 1–12.

Williams, Lance [1983]. "Pyramidal parametrics". In: *Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, pp. 1–11.

Williams, Nathaniel, David Luebke, Jonathan D. Cohen, Michael Kelley, and Brenden Schubert [2003]. "Perceptually Guided Simplification of Lit Textured Meshes". In: *In Proceedings of the 2003 symposium on Interactive 3D graphics*. ACM Press, pp. 113–121.

Witkin, Andrew P. [1987]. "Scale-Space Filtering". In: *Readings in Computer Vision*. Elsevier, pp. 329–332. DOI: https://doi.org/10.1016/B978-0-08-051581-6.50036-2.

Wuerger, Sophie, Maliha Ashraf, Minjung Kim, Jasna Martinovic, María Pérez-Ortiz, and Rafal K. Mantiuk [Apr. 2020]. "Spatio-chromatic contrast sensitivity under mesopic and photopic light levels". In: *Journal of Vision* 20.4, p. 23. ISSN: 1534-7362. DOI: 10.1167/jov.20.4.23. URL: https://jov.arvojournals.org/article.aspx?articleid=2765519.

Yadav, Himanshu and B Annappa [2017]. "Adaptive GPU resource scheduling on virtualized servers in cloud gaming". In: *2017 Conference on Information and Communication Technology (CICT)*. IEEE, pp. 1–6.

Yang, Lei, Dmitry Zhdan, Emmett Kilgariff, Eric B Lum, Yubo Zhang, Matthew Johnson, and Henrik Rydgård [2019]. "Visually Lossless Content and Motion Adaptive Shading in Games". In: *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2.1. NAS, pp. 1–19.

Yin, Lijun, Xiaochen Chen, Yi Sun, Tony Worm, and Michael Reale [2008]. "A High-Resolution 3D Dynamic Facial Expression Database". In: *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 1–6. DOI: 10.1109/AFGR.2008.4813324.

Yin, Lijun, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J. Rosato [2006]. "A 3D Facial Expression Database For Facial Behavior Research". In: *International Conference on Automatic Face and Gesture Recognition*. IEEE, pp. 211–216. DOI: 10.1109/FGR.2006.6.

Zhang, L., Y. Shen, and H. Li [2014]. "VSI: A visual saliency-induced index for perceptual image quality assessment". In: *IEEE Transactions on Image Processing* 23.10, pp. 4270–4281. DOI: 10.1109/TIP.2014.2346028.

Zhang, L., L. Zhang, X. Mou, and D. Zhang [2011]. "FSIM: A feature similarity index for image quality assessment". In: *IEEE Transactions on Image Processing* 20.8, pp. 2378–2386. DOI: 10.1109/TIP.2011.2109730.

Zhang, X. and B. A. Wandell [1997]. "A spatial extension of CIELAB for digital color-image reproduction". In: *Journal of the Society for Information Display* 5.1, p. 61. ISSN: 10710922.

Zhang, Xing, Lijun Yin, Jeffrey F. Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, and Peng Liu [2013]. "A High-Resolution Spontaneous 3D Dynamic Facial Expression Database". In: *International Conference on Automatic Face and Gesture Recognition*. IEEE, pp. 1–6. DOI: 10.1109/FG.2013.6553788.

Zhao, Wenyi, Rama Chellappa, P. Jonathon Phillips, and Azriel Rosenfeld [2003]. "Face Recognition: A Literature Survey". In: *ACM computing surveys* 35.4, pp. 399–458. DOI: 10.1145/954339.954342.

Zhong, Fangcheng, Akshay Jindal, Ali Özgür Yöntem, Param Hanji, Simon J. Watt, and Rafał K. Mantiuk [Dec. 2021]. "Reproducing reality with a high-dynamic-range multi-focal stereo display". In: *ACM Transactions on Graphics* 40.6, pp. 1–14. ISSN: 0730-0301. DOI: 10.1145/3478513.3480513. URL: https://dl.acm.org/doi/10.1145/3478513.3480513.

Zhu, Hao, Haotian Yang, Longwei Guo, Yidi Zhang, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao [2021]. "FaceScape: 3D Facial Dataset and Benchmark for Single-View 3D Face Reconstruction". In: *arXiv preprint*. DOI: 10.48550/arXiv.2111.01082.