Perceptual graphic rendering and quality evaluation

Dong, Lu

2016

Dong, L. (2016). Perceptual graphic rendering and quality evaluation. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/65917>

<https://doi.org/10.32657/10356/65917>

PERCEPTUAL GRAPHIC RENDERING AND QUALITY EVALUATION

DONG LU

School of Computer Engineering

2016

PERCEPTUAL GRAPHIC RENDERING AND QUALITY EVALUATION

DONG LU

School of Computer Engineering

A thesis submitted to Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2016

Acknowledgements

I would like to express my sincere thanks to my supervisor, Dr. Lin Weisi who has continuously encouraged, inspired and supported me throughout the past five years. During my PhD study, I encountered many difficulties in research, Dr. Lin always gave me advices on how to solve the problems. I learned a lot from him and his encouragement really helps me make progress. I also want to thank Dr. Fang Yuming and Dr. Deng Chenwei, for their solid suggestions, scientific ideas and patience. Besides, I want to thank my co-supervisor Dr. Zhu Ce, Dr. Wu Shiqian and Dr. Hock Soon Seah for their guidance, support and suggestions.

I also would like to thank the members in our group, Dr. Liu Anmin, Dr. Manish Narwaria, Dr. Gu Zhouye, Dr. Yang Huan, Mr. Nevrez Imamoglu, Dr. Zhang Lining, Ms. Li Qiaohong, Dr. Wu Jinjian, Mr. Cheng Wentao, Mr. Yuan Yuan, Mr. Prakhya Sai Manoj, Mr. Zhang Yabin, and Mr. Sharath Chandra Guntuku. Discussions with them were really fruitful and pleasant. Without them, the project cannot be such an enriching and enjoyable experience.

Last, many thanks to my good friends, Ms. Cheng Yue, Ms. Cheng Peng, and Ms. Yuan Zhen for their friendship and support.

Abstract

Computer graphics are widely used in various areas nowadays, such as entertainment, industry design, education, architecture and scientific visualization. Graphic rendering is the process of displaying 3D models visually as 2D images. An important issue in graphic rendering is to improve the rendering efficiency. Since the ultimate evaluator of the realistically rendered images is the human visual system (HVS), it is meaningful to substantially decrease computational, storage and transmission cost by exploiting perceptual properties of the HVS without degradation of perceived rendering quality.

To design a perceptual rendering system, we need to consider two processes: one is how to build computational models of the characteristics of the HVS; the other is how to incorporate these models into the rendering process. In this work, we explore both the two processes for three schemes in graphic rendering. Firstly, we propose a new visual saliency based perceptual rendering scheme. In general, a scene has one or more salient objects, and the visual attention of observers is attracted by the salient objects. This property of the HVS allows us to decrease computational resources in non-salient objects without degrading the perceived rendering quality. Different from existing work in saliency-based perceptual rendering which use visual saliency as the

only guide for computation allocation, we integrate rendering complexity with visual saliency to drive rendering. The rendering complexity of each pixel is estimated with a sample variance metric, and the computational resources are iteratively distributed among pixels according to visual saliency and rendering complexity.

Secondly, we build an enhanced visual saliency detection method for rendered graphics by making use of 3D model information. The proposed method computes the object-level contrast between each object and the remaining objects in a rendered image, and detects salient objects based on the graphic contrast. The proposed method is able to find the accurate boundaries of salient objects and improve the accuracy of saliency detection. We also incorporate this enhanced saliency detection method in our saliency-based perceptual rendering scheme, to generate consistent rendering quality in salient objects.

Thirdly, we explore a property of the HVS that has not been exploited by existing rendering applications, i.e. low sensitivity to irregular structure. Some 3D scenes include a large amount of irregular textural information, and the noise masking ability of irregular structure allows us to decrease the computational resources in irregular structure, without introducing visible artifacts into the rendered image. We propose an irregularity measure which relies on the similarity between each pixel and the neighboring pixels. The irregularity measure is applied in a perceptual rendering scheme to improve the rendering efficiency of scenes with a large amount of irregular texture.

In addition, we also explore perceptual quality evaluation of 3D models. Perceptual quality evaluation of 3D models is critical for design and optimization of related graphic algorithms, such as model simplification, compression and perceptual

rendering. 3D models are widely represented by 3D triangle meshes, due to their simplicity. We design a novel objective quality evaluation method to assess the quality of distorted 3D meshes with respect to a reference one. Curvature difference between each pair of corresponding vertices is calculated, and then modulated by two new components, namely the visual masking module and the saturation effect module. Structural distortion between 3D meshes is also calculated and pooled with the said curvature difference into a quality score. The proposed quality evaluation method yields consistent results in three publicly available databases.

Contents

Acknowledgements	ii
Abstract	iv
Contents	vii
List of Figures	x
List of Tables	xiv
List of Abbreviations	xvi
1 Introduction	1
1.1 Background, Motivation and Overview	1
1.2 Objective and Scope of this Work	8
1.3 Major Technical Contributions	8
1.4 Organization of the Thesis	10
2 General Literature Survey	12
2.1 Adaptive Sampling for Graphic Rendering	12

2.2	Perceptual Graphic Rendering Techniques	16
2.3	Quality Evaluation of 3D Models	21
3	Perceptual Rendering Based on Visual Saliency	24
3.1	Introduction	25
3.2	Saliency-based Perceptual Rendering	29
3.2.1	Saliency Detection with the Phase of Fourier Transform (PFT)	31
3.2.2	Complexity Estimation in Rendering	32
3.3	Experimental Results and Discussions	36
3.3.1	Comparison of Saliency-based Perceptual Rendering	36
3.3.2	Cost Reduction Measurement	42
3.4	Discussions	47
3.5	Concluding Remarks	50
4	Enhanced Visual Saliency Detection for Rendered Graphics	52
4.1	Introduction	53
4.2	Enhanced Visual Saliency Concept and Detection	57
4.2.1	Dominant Color Extraction	59
4.2.2	Feature Channel Extraction for Each Object	61
4.2.3	Computation of Object-level Contrast	62
4.2.4	Iterative Suppression	65
4.2.5	Discussion on Object-level Contrast Computation	67
	4.2.5.1 Comparison between Object-level Contrast and Multi- scale Pixel-level Contrast	67

4.2.5.2	Comparison between Global Contrast and Local Contrast	69
4.3	Experimental Results and Discussions	71
4.4	Application in Perceptual Rendering	77
4.5	Concluding Remarks	81
5	Perceptual Rendering with Irregularity Measure	83
5.1	Introduction	84
5.2	Perceptual Rendering Based on Irregularity Measure	88
5.2.1	Existing Entropy Masking Algorithm	88
5.2.2	The Proposed Irregularity Measure	89
5.2.3	Perceptually Based Adaptive Sampling	96
5.2.4	Discussion for Irregularity Measure Parameters	99
5.3	Experimental Results and Discussions	102
5.3.1	Comparison to Existing Visual Masking Methods in Perceptual Rendering	102
5.3.2	Speedup Measurement	106
5.4	Concluding Remarks	113
6	Perceptual Quality Assessment for 3D Meshes	115
6.1	Introduction	116
6.2	The Proposed Quality Evaluation Method	120
6.2.1	Overview	120
6.2.2	Computation of Mean Curvature Difference	121
6.2.3	Visual Masking Modulation	124

6.2.4	Saturation Effect Modulation	128
6.2.5	Structural Distortion	130
6.3	Experimental Results and Discussions	133
6.3.1	Databases	134
6.3.2	Accuracy and Robustness Evaluation	135
6.3.3	Impact of Parameters	139
6.4	Concluding Remarks	143
7	Summary and Future Work	145
7.1	Summary	145
7.1.1	Incorporation of Visual Saliency in Perceptual Rendering . . .	146
7.1.2	Incorporation of Visual Masking in Perceptual Rendering . . .	148
7.1.3	Quality Evaluation for 3D Meshes	149
7.2	Limitations	150
7.3	Future Work	151
7.3.1	Integration of More Visual Perceptual Characteristics	152
7.3.2	Quality Evaluation of 3D Models and Rendered Graphics . . .	154
	Bibliography	157
	List of Published Work	177

List of Figures

1.1	Major contents and organization of the thesis.	11
2.1	Illustration of ray tracing algorithm.	13
2.2	Quantities used in the rendering equation.	14
3.1	An example showing that increasing the sampling rate at salient pixel is not always helpful in quality enhancement	27
3.2	An illustration of large distortion turning a non-salient region to a salient one	28
3.3	Block diagram of the proposed adaptive sampling scheme.	30
3.4	Saliency map of the Cornell scene	32
3.5	Error values as estimation of rendering complexity	34
3.6	The priority map generated by the proposed method.	35
3.7	Comparison of images rendered perceptually with the saliency map and the priority map (I)	37
3.8	Saliency map and priority map for the Statues scene	38
3.9	Comparison of images rendered perceptually with the saliency map and the priority map (II)	39

3.10	Comparison of images rendered perceptually with the saliency map and the priority map (III)	40
3.11	Reference image and test image for the Dragons scene	47
3.12	Comparison of saliency detection results using different models	48
3.13	Comparison of perceptual rendering results using different saliency de- tection models	49
4.1	The problem of existing visual saliency detection schemes.	56
4.2	The framework of the proposed visual saliency detection method. . .	58
4.3	Example of dominant color extraction results.	61
4.4	Saliency detection results with different T_c and ξ values.	64
4.5	Saliency maps from different operators.	66
4.6	Illustration of the multi-scale contrast computation.	67
4.7	Comparison of saliency maps computed from multi-scale contrast and object-level contrast.	69
4.8	Comparison between saliency maps computed based on local object contrast and global object contrast.	70
4.9	Saliency maps from different saliency detection methods.	72
4.10	Experiment results for the comparison between the proposed and other methods.	75
4.11	Perceptually rendered images for the Tribal scene by using different saliency detection methods.	78
4.12	Perceptually rendered images for the Fairy scene by using different saliency detection methods.	79

5.1	Examples of masking properties of different image contents.	85
5.2	The pipeline of the proposed irregularity detection scheme.	90
5.3	Filtering results of two patterns using the luminance difference and the patch difference.	91
5.4	Filtering result of one regular pattern using the patch difference. . . .	94
5.5	Irregularity detection result of the Room scene.	95
5.6	Block diagram of the proposed adaptive sampling scheme in path tracing.	97
5.7	The effect of different γ in sample distribution of the Room scene. . .	98
5.8	Comparison of different patch sizes.	100
5.9	Computational cost of the filtering step on the Teapot scene with dif- ferent neighborhood sizes.	101
5.10	Filtered images of the Tribal scene using different ς	102
5.11	Comparison of the irregularity maps by the proposed method and the visual masking maps by existing visual masking computation methods.	103
5.12	Comparison to existing visual masking schemes in perceptual image rendering.	105
5.13	Test scenes used in the experiment.	106
5.14	HDR-VDP-2 visual differences between the reference images and the test images by the proposed method.	110
6.1	The pipeline of the proposed 3D mesh quality assessment method. . .	120
6.2	Mean curvature describes the visual characteristics of 3D models. . .	122
6.3	Quantities used in computation of curvature tensor.	123
6.4	An example of visual masking effect.	124

6.5	Mean curvature maps for the LionVase model.	126
6.6	Scatter plots of the MOSs versus the curvature difference for the Lion- Vase model.	128
6.7	Example on saturation effect.	129
6.8	Saturation effect modulation function.	130
6.9	Noise distorted and smoothing distorted models of the Venus.	131
6.10	Scatter plots of MOSs versus predicted score by the proposed method.	132
6.11	Maximum curvature map of the Venus model.	132
6.12	Scatter plots of MOSs versus the predicted scores from the proposed model for the meshes in the three databases.	136
6.13	Plot of r_p as a function of σ	140
6.14	Impact of β on the performance.	141
6.15	Plot of r_p as a function of T_r	142
6.16	Plot of r_p as a function of α	143

List of Tables

3.1	Comparison of sampling rates and rendering quality of non-salient regions (I)	40
3.2	Comparison of sampling rates and rendering quality of non-salient regions (II)	41
3.3	The numbers of viewers in favor of images rendered by the two algorithms.	44
3.4	Output for a χ^2 test	45
3.5	Cost for the reference images and test images by the proposed method.	46
4.1	Cost of the saliency detection methods	76
5.1	The numbers of viewers in favor of images rendered by the two algorithms.	108
5.2	Output for a χ^2 test	109
5.3	Cost for the reference images and test images by the proposed method.	112
6.1	MOSs, curvature difference, and modulated curvature difference of two distorted models	127
6.2	Correlation coefficients of different objective metrics on LIRIS/EPFL general-purpose database.	137

6.3	Correlation coefficients of different objective metrics on LIRIS masking database.	138
6.4	Correlation coefficients of different objective metrics on UWB compression database.	139

List of Abbreviations

2AFC	Two Alternatives Forced Choice
CSF	Contrast Sensitivity Function
DCD	Dominant Color Descriptor
HD	Hausdorff Distance
HDR VDP	High Dynamic Range VDP
HVS	Human Visual System
JND	Just Noticeable Distortion
LoD	Level of Detail
MOS	Mean Opinion Score
MSE	Mean Square Error
PBRT	Physically Based Rendering Technology
PFT	Phase of Fourier Transform
PSNR	Peak Signal Noise Ratio
RMS	Root Mean Square Error
TVI	Threshold Vs Intensity
VDM	Visual Discrimination Model
VDP	Visual Difference Predictor

VE	Virtual Environment
VEP	Visual Equivalence Predictor

Chapter 1

Introduction

1.1 Background, Motivation and Overview

Computer graphics are increasingly important to the growing requirement of entertainment, industry design, education, architecture and scientific visualization. A 3D model is a mathematical representation of geometrical data and can be displayed visually as 2D images through a rendering process. Among various subareas in computer graphics, graphic rendering, the process of visualizing 3D models as 2D images, plays an important role. A graphic rendering system takes a scene file, which includes 3D models (textured or non-textured), and lighting, viewing and shading information, as an input, and generates a 2D image by simulating the light transport in the 3D virtual scene [41].

With the rapid development of rendering algorithms and graphics hardware, high-quality realistic renderings can be created. However, synthesizing high-quality realistic images is very time-consuming, especially when object models are sophisticated

and accurate light simulation is required [103]. Therefore, how to decrease the computational cost becomes an important research issue in graphic rendering, especially for green engineering and in mobile computing (where processing power and battery life are critical).

To accelerate rendering from the aspect of graphics hardware, graphics processing unit (GPU) has been used. A CPU consists of a few cores optimized for sequential serial processing while a GPU has a massively parallel architecture consisting of thousands of cores, thus a GPU is able to handle multiple tasks simultaneously. Various methods, such as [38, 100], have been proposed to adapt CPU-based global illumination methods to the GPU. Global illumination methods, such as ray tracing, trace light rays from image plane to the scene, and simulate the physical interaction between light rays and object surfaces. The computation of global illumination consists of random/non-sequential memory access and irregular distribution of workload, thus it cannot be easily handled by the GPU parallel computation.

To accelerate rendering from the aspect of rendering algorithms, perceptual based rendering has been proposed. Perceptual rendering systems take characteristics of the HVS (human visual system) into account. Since the ultimate evaluator of most (if not all) of the realistically rendered images is the HVS, it is possible to reduce rendering cost by exploiting perceptual properties of the HVS without significant loss in perceived rendering quality. As we know, the processing resources for information are limited in the HVS, and thus instead of receiving and processing all the visual stimuli, the HVS tends to focus on the most relevant visual information [11]. Some visual information is ignored by the HVS and such characteristics allow the reduction of computational cost without degrading the perceived quality of rendered graphics [6].

Visual attention mechanism is an intensively studied characteristic of the HVS. The stimulus itself can provide the guide, i.e. the attention can be attracted involuntarily by salient visual stimuli [141]. Visual saliency information is useful in applications where the rendering process is a passive process, e.g. game and virtual tour. In applications where the rendering process is an active process, e.g. animation, the artist takes full control of the image to be rendered, and saliency computation may not be necessary since the artist knows what needs to be rendered in detail. However, in passive rendering applications, a user can interact with the scene, and images are rendered according to the virtual viewpoint of the user. Since salient objects attract more attention than non-salient objects, the overall rendering cost can be decreased by directing less computational resources to non-salient objects, without degradation of perceived rendering quality. There are some existing perceptual rendering schemes based on the visual saliency, such as [14, 52, 68, 69, 77, 148]. These schemes first detect the saliency levels of different regions in a coarsely-rendered input image, usually by the use of a saliency detection model (such as the one in [48]), and then allocate less computational resources to regions with lower saliency levels in the rendering process. These schemes use the saliency level as the only guide for computation allocation, however, the rendering complexity is not considered; computational resources may be wasted on low-complexity regions, while artifacts may be introduced to high-complexity regions.

To address the problems, in the work documented in this thesis we explore how to incorporate visual saliency in the rendering process seamlessly to reduce rendering cost, and propose a new saliency-based perceptual rendering scheme. In pre-processing stage of rendering, a coarse image is rendered and the saliency level of

each pixel is detected by a saliency detection model. Any visual saliency detection method can be incorporated in the proposed scheme, and we use a PFT (Phase of Fourier Transform) based saliency detection method [34] as an example. In rendering, the rendering complexity of each pixel is estimated with a sample variance metric, to reflect the computational cost for rendering each pixel. The complexity value is integrated with the saliency value into a priority value, and the rendering resources are iteratively distributed among pixels according to priority values, until the computational budget exhausts. Compared with perceptual rendering that is merely based upon visual saliency, the proposed method is able to yield better rendering quality, under the limited computational resources.

In the area of computer vision and image/video processing, many computational methods have been devised to detect salient regions for various visual signal processing applications, such as object detection [95], object tracking [98], and image retargeting [28, 35]. These methods extract features from 2D images and compute saliency as pixel-level contrast between features of the center pixels and the neighboring pixels: pixels with larger contrast to their surroundings are assigned higher saliency values. Due to the lack of scene information, it is difficult to detect the whole salient objects by saliency detection methods for natural images. Several saliency detection methods for rendered graphics try to detect salient regions by using 3D model information [43, 68, 77], but these methods still rely on pixel-level contrast and can not detect accurate boundaries of salient objects.

In the context of perceptual rendering, inaccurate saliency detection results leads to quality inconsistency within a single object in a rendered image. For example, if some pixels of an object are detected as salient ones while the remaining pixels

are detected as non-salient ones, salient pixels of the object are rendered with high quality and non-salient pixels are rendered with low quality. Since the HVS perceives each salient object as a whole [49, 94, 114, 143], the quality change within an object is noticeable to the HVS and degrades the overall quality of the rendered image.

To improve the saliency detection accuracy as well as to solve the above mentioned problem in saliency-based perceptual rendering, we propose an enhanced visual saliency detection method for rendered graphics. First, the accurate object segmentation information is extracted from rendered graphics, and each segmented object is represented by a Dominant Color Descriptor (DCD). The object-level contrast in DCD between each object and the remaining objects in the image is calculated, and objects with high contrast values are detected as salient ones. We compare the performance of the enhanced saliency detection method with the relevant existing methods [34, 43, 47, 68]. Experimental results demonstrate that the enhanced method is able to detect the accurate boundaries of salient objects and improves the accuracy of saliency detection. We also incorporate the enhanced visual saliency detection method in perceptual rendering.

Besides the visual saliency, we also explore a property of the HVS that has not been exploited in rendering application in the literature, i.e. low sensitivity to irregular structure. According to the concept of entropy masking proposed by Watson *et al.* [138], visibility of visual signals (e.g. noise) decreases due to the presence of a mask signal which is unknown to human eyes. Watson *et al.* [138] also indicated that the degree to which a pattern is unknown is related with the complexity of the pattern, i.e. a complex pattern has strong masking effect than a simple pattern. An irregular pattern is with high complexity and thus has strong visual masking effect. In both

active rendering processes, e.g. animation, and passive rendering processes, e.g. game and virtual tour, an irregularity measure of the pixel can be exploited in rendering acceleration for graphics which contains a large amount of irregular structure. the overall rendering cost can be decreased by directing less computational resources to pixels with strong masking effect, without degradation of perceived rendering quality.

We propose an irregularity measure for rendered graphics. A filter is designed to calculate the weighted average of each pixel’s neighboring pixels. After filtering, structural information is preserved while irregular details are removed. Based on the difference between the original and filtered images, irregular regions can be well distinguished from regular ones. We incorporate the irregularity measure in a rendering scheme to improve rendering efficiency by distributing less computational resources to irregular image regions. A coarse image is rendered, and the irregularity level of each pixel is estimated with the proposed method. In the perceptual rendering process, rendering complexity of each pixel is also estimated and integrated with the irregularity value into a priority value. Computational resources are iteratively distributed among pixels accordingly. The proposed perceptual rendering scheme is especially suitable for rendered graphics with a large amount of irregular texture, since the potential reduction in rendering computation is large.

Level of details (LoD) serves a similar purpose as the proposed perceptual rendering methods in the thesis. The basic idea of LoD is using a simplified 3D model for rendering of small, distant or non-salient objects in a scene [82]. In a LoD scheme, each object is represented by several levels of simplified versions. As the level increases, the model becomes simpler and faster to render. A LoD scheme decreases the rendering cost by modulating the geometry complexity in the 3D domain. Different

from the LoD scheme, the proposed perceptual rendering methods in the thesis are implemented by modulating variables in the image domain, i.e. the number of sample rays traced per pixel, according to the perceptual properties of the HVS. In fact, the proposed saliency detection model and the proposed irregularity measure can also be incorporated in a LoD scheme.

Besides perceptual graphic rendering, we also explore objective quality evaluation of 3D models which are represented as triangle meshes. Visual quality assessment of 3D meshes is critical since such a criterion can be used in design and optimization of graphic related algorithms and systems. A quality metric for 3D meshes is also useful for the purpose of perceptual rendering. Traditionally, the visual quality is evaluated by geometric distance such as Hausdorff Distance (HD) and Root Mean Square Error (RMS); however, these metrics do not correlate well with the human judgement [19]. To design an objective quality metric which correlates well with the human judgement, properties of the HVS should be considered [19].

In the relevant part of the documented work, we design a novel perceptual quality evaluation method for 3D meshes. Given a reference mesh and a distorted mesh, the curvature difference between each pair of vertices is calculated. The curvature difference is then modulated by the visual masking module and the saturation effect module, to account for the properties of the HVS. Structural distortion between the two 3D meshes is also calculated. Finally, the structural distortion is integrated with the average curvature difference into a quality score.

1.2 Objective and Scope of this Work

The objective of this study is to explore properties of the HVS and incorporate the related modeling in graphic rendering and quality evaluation.

In particular, we focus on the following four problems in graphic rendering and quality evaluation. First, how to incorporate a visual saliency detection model in graphic rendering process seamlessly to improve the rendering efficiency? Second, how to build a saliency detection model for rendered graphics by making proper use of 3D model information? Third, is there any other perceptual properties which can be exploited in rendering besides those that have been studied and exploited? Last, how to build an objective quality evaluation method for 3D models by exploiting perceptual properties of the HVS?

1.3 Major Technical Contributions

The key technical contributions of this research can be summarized as follows:

- We have proposed a new visual saliency based perceptual rendering scheme. Visual saliency is usually determined by computational visual attention models. Existing saliency-based perceptual rendering methods use saliency level as the only guide for allocation of computational resources. We have first demonstrated that using saliency level as the only guide will introduce artifacts in the rendered images. Then we have exploited the rendering complexity and proposed a perceptual rendering scheme by allocating computational resources according to both the saliency level and the estimated complexity level.

- We have proposed an enhanced visual saliency detection scheme for rendered graphics by making proper use of 3D model information. Different from existing saliency detection methods, which estimate saliency based on pixel-level contrast, the proposed method detects salient objects by computing object-level contrast. Given a rendered image, the proposed method first extracts accurate object segmentation information from the 3D models. The saliency of each object is then calculated by measuring the contrast between the features of the object and its surrounding objects. The proposed saliency detection scheme is incorporated in perceptual rendering to demonstrate its advantages.
- We have proposed an irregularity measure and incorporated this measure in perceptual rendering application. We have introduced the low sensitivity to irregular structure in the HVS and then devised a method to detect the irregularity level of image regions. We have incorporated the irregularity measure in perceptual rendering by distributing computational resources according to the irregularity level and the estimated rendering complexity of image pixels.
- We have designed a novel objective quality assessment method for 3D meshes. We have proposed a novel objective quality assessment method for 3D meshes, based on curvature information. According to the characteristics of the HVS, two new components including the visual masking module and saturation effect module have been designed for the proposed method.

1.4 Organization of the Thesis

Figure 1.1 illustrates the major contents and organization of this thesis. The thesis has been divided into seven chapters as described as follows.

Chapter 1 (this chapter) gives a brief introduction about the thesis, including the background motivation, and overview of objective and scope, thesis contributions, and thesis organization.

Chapter 2 introduces the major related existing work and algorithms in adaptive sampling for graphic rendering, perceptual graphic rendering, and objective quality evaluation of 3D models. More specific literature survey to each proposed technique in this thesis will be further introduced whenever appropriate in Chapters 3 - 6.

Chapter 3 discusses visual saliency based perceptual rendering. The problems in perceptual rendering which is merely based upon visual saliency are illustrated. A complexity metric in rendering is described, and integrated with the saliency metric to drive perceptual rendering.

Chapter 4 describes an enhanced saliency detection method for rendered graphics using 3D model information, as well as the application in graphic rendering. Accurate object segmentation information, which is generally not available in 2D natural images, is extracted from 3D scenes and used in the proposed saliency detection method. The proposed saliency detection method computes an object-level saliency with the object segmentation information.

Chapter 5 focuses on an irregularity measure and its application in a perceptual rendering scheme. The irregularity measure describes the noise masking potential of image pixels due to the structural irregularity. Less computational resources are

Figure 1.1: Major contents and organization of the thesis.

distributed to pixels with higher irregularity levels and the overall computational cost decreases without degrading the perceived quality of the rendered images.

Chapter 6 introduces a proposed visual quality evaluation method for 3D meshes. The proposed method relies on the curvature comparison between a reference 3D mesh and a distorted 3D mesh. The per-vertex curvature difference is modulated by a visual masking modulation module and a saturation effect modulation module, to account for the relevant properties of the HVS.

Chapter 7 concludes the thesis with a summary of the main research work performed and the possible directions of further studies.

Chapter 2

General Literature Survey

This chapter gives a general literature review on adaptive sampling for graphic rendering, perceptual graphic rendering, and objective quality evaluation of 3D models, while more specific review toward the performed research will be presented at the beginning of Chapters 3 to 6, respectively.

2.1 Adaptive Sampling for Graphic Rendering

In graphic rendering, ray tracing is a technique for generating an image by tracing light rays into a virtual scene and simulating the interaction between rays and virtual objects [103]. Ray-tracing based graphic rendering algorithms are widely used to generate high-quality realistic images by computing the global illumination [103]. Such algorithms include path tracing [54], bidirectional path tracing [61], photon mapping [50], irradiance cache [59], metropolis light transport [132], and so on.

Figure 2.1 illustrates the ray tracing process: for each pixel in the virtual image plane, a number of rays are traced from a virtual camera to sample the scene. These

Figure 2.1: Illustration of ray tracing algorithm.

light rays intersect with objects in a scene and hit on the virtual light sources after multiple reflections or refractions. The radiance returned by each light ray is calculated by solving a rendering equation [54], which is shown in Equation 2.1.1. The quantities used in the rendering equation are illustrated in Figure 2.2.

$$Ra_{out}(o, \tau_r) = Ra_{emit}(o, \tau_r) + \int_{Hemi} Ra_{in}(o, \tau_i) \cdot Brdf(o, \tau_r, \tau_i) \cdot \cos(\psi_i) \cdot d\tau_i \quad (2.1.1)$$

where, $Ra_{out}(o, \tau_r)$ is the radiance returned by a light ray; τ_r is the solid angle around the outgoing light ray; $Ra_{emit}(o, \tau_r)$ is the radiance emitting from point o on the surface; $Ra_{in}(o, \tau_i)$ is the incoming radiance to the surface; τ_i is the solid angle around the incoming light ray; $Brdf(o, \tau_r, \tau_i)$ is the bidirectional reflectance distribution function, which defines how light is reflected at an opaque surface; ψ_i is the angle between the incoming ray and surface normal; $d\tau_i$ is the differential solid angle around the incoming ray. The radiance $Ra_{out}(o, \tau_r)$ returned by the light ray is the sum of two components: the emitted radiance from the intersection point and the integral of the incoming radiance over the hemisphere (Hemi) of the incoming rays.

The integral is evaluated with numerical estimation, such as the Monte Carlo

Figure 2.2: Quantities used in the rendering equation.

algorithm [60, 117]. The Monte Carlo estimator is expressed as:

$$Est = \frac{1}{N} \sum_{i=1}^N \frac{Ra_{in}(o, \tau_i) \cdot Brdf(o, \tau_r, \tau_i) \cdot \cos(\psi_i)}{p(\tau_i)} \quad (2.1.2)$$

where N is the number of light rays; $p(\tau_i)$ is the distribution of incoming sample rays. The radiance returned by discrete samples are then used to reconstruct a new image function similar to the original image function. The discrete pixel values in rendered digital images are calculated from the reconstructed image function. Since the sampling and reconstruction processes include approximation processing, errors will be introduced into the rendered images. A large number of samples (i.e. large N) are usually required for each pixel to avoid noise appearing in resultant images, which increases the computation burden.

Adaptive sampling schemes are introduced to reduce the number of samples computed for ray-tracing based rendering algorithms, by concentrating samples on image regions with higher rendering complexity in an iterative way [124].

The pioneer work in [140] uses contrast to guide adaptive sampling. Each pixel is defined as a rectangle cell, and one sample ray is traced at each of the four corners of the cell. If the contrast of the sample values is large, the cell is further divided

until the contrast becomes sufficiently small. The contrast metric is further developed in [87] by taking into consideration of characteristics of the HVS. Dark image regions are given larger weights in the contrast metric, but of course, this is just an overly simple case for the HVS consideration.

The method in [99] detects high complexity regions in the wavelet domain. Intermediate rendering results are represented in the wavelet domain, and the variance of the wavelet coefficients is calculated to detect image regions with high variance either in image domain or in other dimensions (such as time dimension for motion blur effect and lens dimensions for out of focus effect).

The work in [37] uses the contrast metric in [87] to locate high contrast regions in image and other domains. It generalizes the contrast metric in the multidimensional space. The contrast is calculated using samples located within the high dimensional neighborhood instead of image domain neighborhood. With the multidimensional contrast metric, high-contrast graphic contents (e.g. edge, motion blur, out of focus) can be detected more directly and accurately than using only the image domain contrast metric as in [99].

Apart from the image-domain adaptive sampling, frequency analysis has been performed on rendering distributed effects, such as motion blur, soft shadow and depth of field, in [26], [25] and [119], respectively. Based on the frequency analysis, adaptive sampling schemes have been proposed to render each specific effect. The work in [70] does not perform frequency analysis, but it also uses the sheared reconstruction filters inspired by the frequency analysis in [25]. Different from the methods in [25, 26, 119], the method in [70] can render the motion blur, soft shadows and out of focus effects simultaneously rather than a specific effect. Rousselle *et al.* [111, 112] attempted to

minimize MSE (mean square error) by adaptively filtering out the high frequency noise. Kalantari *et al.* [55] used both image-space adaptive sampling and adaptive filters in their scheme.

In existing adaptive sampling schemes, the properties of the HVS have rarely been considered (with a few exceptions, e.g. the method of [87] which considers the luminance adaptation of the HVS). Therefore, in the work documented in this thesis, we explore the possibility to integrate the properties of the HVS and the adaptive sampling for further reduction of computational cost.

2.2 Perceptual Graphic Rendering Techniques

Perceptual rendering tries to decrease rendering cost by modeling and incorporating properties of the visual perception. Comprehensive reviews of the perceptual graphic rendering techniques can be referred to [6, 96, 126].

The human visual attention mechanism is one of the most intensively studied topics in psychology and cognitive neuroscience, and it is also referred to as the allocation of processing resources [3]. Since the human’s processing resources are limited, the HVS tends to selectively concentrate on one aspect of their surrounding environment while ignoring the other aspects. In general, the visual attention mechanism is thought to operate as a two-stage process [53]. In the first stage, attention is allocated uniformly over the visual image. In the second stage, attention is concentrated to certain region of the image for in-depth information processing. There are two mechanisms in visual attention. The first one is bottom-up processing, which does not depend on the observer’s knowledge of the stimulus. The other one is called

top-down processing, which depends on the task and the observer’s knowledge [141], i.e. the attention is attracted voluntarily.

Visual saliency based perceptual rendering algorithms distribute more computational resources to image regions where the observers pay more attention to, and less computational resources to the remaining regions. The work by Yee *et al.* [144] was the first attempt to exploit visual attention in rendering acceleration. A coarse image is rendered as a quick estimation of the final image, using fast GPU-based methods. Visible error threshold, i.e. the error level to which the HVS can tolerate, of each pixel in the coarse image is calculated. The estimated error threshold is weighted by the detected saliency level to guide rendering. By allocating less computational resources to regions with higher error threshold levels, the computational cost for indirect lighting in radiance caching algorithm decreases.

Haber *et al.* [36] incorporated visual saliency in an interactive rendering system. They improved the saliency detection model in [144], by making use of the 3D object information such as objects’ distance from image center and size of each object. A coarse image is rendered using graphics hardware and only the appearance of non-diffuse objects that are selected by the saliency detection model is corrected.

Longhurst *et al.* [76, 77, 78] exploited visual saliency in ray tracing. They proposed a saliency detection model for rendered graphics by using model information including motion, depth, and habituation. In their scheme, sampling rate for each pixel is determined by the saliency level, and a large number of sample rays are distributed to pixels with high saliency levels.

Lee *et al.* [68, 69] considered visual saliency in interactive rendering of virtual environment (VE). They built a saliency detection model for rendered graphics by using

image-domain feature as well as 3D model information including object segmentation, depth and size.

Hasic *et al.* [40] conducted psychophysical experiments to investigate the perception of high-saliency movement of objects in a dynamic scene. The experimental results show that the rendering cost can be decreased when motion is introduced to static objects, without any perceptual loss of quality.

Besides bottom-up mechanism, top-down attention mechanism has also been studied in saliency-based perceptual rendering. Cater *et al.* [12, 13] incorporated a top-down attention model in ray tracing. Viewers are given a visual task, e.g. to find out how many teapots exist in a rendered image, and then a task map is generated to predict where people will focus their attention to the given specific task, following the principle of inattention blindness. Guided by the task map, more samples are distributed to pixels with higher saliency levels in the rendering.

Sundstedt *et al.* [122] combined bottom-up and top-down mechanisms in a ray tracing algorithm. A saliency map is generated by a bottom-up visual attention model [48], and a task map is generated from the task information. The two maps are combined into an importance map to guide sample allocation in ray tracing. Later, they extended the use of importance map to perceptual rendering of participating media [4, 123].

Hillaire *et al.* [43] combined bottom-up and top-down mechanisms in interactive rendering of VE. Task information and camera motion information are used together with the bottom-up visual attention model to predict the eye fixations. The results of gaze prediction are used to guide rendering of objects with different LoDs in interactive rendering.

Memory schema, which is a knowledge structure related with the human memory, is studied in [7, 148]. The mechanism indicates the context of a scene and the correlation between objects and the scene. In perceptual rendering schemes of [7, 148], inconsistent objects which are not expected to be found in a scene are determined as attended, and then models of higher LoD are used for rendering.

Apart from the visual attention mechanism, visual metrics based on computational modeling of the HVS are also widely used in perceptual rendering [9, 10, 24, 29, 31, 80, 88, 89, 91, 107]. A visual metric, such as the metrics in [20, 79], takes two images as input, and predicts the visible differences between the two images. Low-level visual perception of the HVS is modeled in the visual metrics, including the luminance adaptation [107], contrast sensitivity function (CSF) [24, 88, 89, 91, 107], and contrast masking [31, 88] etc. In a visual metric based perceptual rendering scheme, a visual metric is utilized to estimate the levels of visible error in intermediate rendering results, and guide the allocation of computational resources adaptively according to the levels of visible error. The rendering stops when the estimated errors are below a predefined error threshold value.

Ramanarayanan *et al.* [105, 106] proposed a visual metric named as “visual equivalence”, which tries to determine whether two rendered images express the same perception information including shape, material and illumination instead of determining whether two images is visually the same. They applied the visual metric in lightcuts [134] and precomputed radiance transfer [118] rendering algorithms, and demonstrated that illumination computation of less accuracy is able to generate images of the same level perceptual quality as the reference images. Later in [58], Ramanarayanan applied the visual metric to control the approximation accuracy of

the virtual point light method.

Besides modeling visual metric, researchers have also worked on perceptual rendering algorithms which take advantage of other aspects of the visual processing. Stokes [121] took into account the perception of illumination components in global illumination rendering. The global illumination computation is decomposed into several direct and indirect illumination components, and illumination components which have lower perceptual importance in the final image quality are rendered with less computational resources. Lo *et al.*[75] studied the human binocular fusing ability for stereoscopic images rendering. In rendering of stereoscopic images, a left-eye view and a right-eye view need to be generated. Lo *et al.*[75] exploited the human binocular fusing ability by decreasing the resolution of one of the two images, without affecting the perceived quality of the synthesized stereoscopic image. Navarro *et al.* [92] conducted psychophysical experiments to investigate the perception of motion blur in image rendering. They demonstrated that the computational cost of motion blur effect can be reduced by exploiting the limits of visual perception in motion blur. This property has been further studied and used in rendering acceleration [116, 145].

Besides the visual perception, cross-modal perception and interaction have also been studied and exploited in perceptual rendering, such as the integration of auditory-visual stimulus [15, 46, 86, 109] and the integration of visual-olfactory stimulus [39, 108].

2.3 Quality Evaluation of 3D Models

3D models are the major source of information in computer graphics, and subject to various processing operations which include compression, simplification, watermarking and so on. Distortions are introduced to 3D models in processing. To predict the human judgement on quality of 3D models, objective quality assessment methods have been designed. The objective quality evaluation methods can be classified into two main categories: image-based and model-based [19]. Detailed reviews of visual metrics can be referred to [19, 65]. Model-based methods extract features from 3D models, and predict the quality of 3D models based on the difference in features between reference models and distorted models. According to results from existing studies, the classical geometric distances, like Hausdorff Distance (HD) and RMS, do not correlate well with human visual perception [19]. Thus perceptually based metrics have been introduced to evaluate the quality of 3D models.

Karni *et al.* [56] argued that a geometrical Laplacian operator captures visual properties of surfaces, e.g. smoothness. They formulated the difference between a reference model and a distorted model as the average of differences in both vertex coordinates and Laplacian values between corresponding vertices in the models. Later, Sorkine *et al.* [120] improved the method [56] by giving a greater weight to the Laplacian values in the distortion measure.

Bian *et al.* [8] proposed a quality assessment metric based on the level of energy which causes the deformation to the reference model, i.e. the strain energy. The metric is able to measure distortion introduced by operations including watermarking, compression and 3D filtering.

Corsini *et al.* [18] proposed a method based on the roughness, which is a measure of the variation in surface vertices. Two formulations of roughness measure were derived: The first one is based on the variation of dihedral angle, which is the angle between two triangles, and the second one is based on the differences between a reference model and a smoothed version of the model. With the roughness measures, the quality of a distorted model is measured based on the per-vertex difference in roughness between the distorted and reference models. Dihedral angle is also explored by Váša *et al.* [131]. The method [131] computes differences in oriented dihedral angles between vertices on 3D models, and pools the per-vertex difference into a global quality score for the distorted 3D model. This metric yields good performance for evaluation of the compression distortion.

Some existing methods [64, 66, 128, 129, 136] use the curvature information extracted from 3D models. Inspired by the SSIM metric (structural similarity index metric) [137] for images, Lavoué *et al.* [66] proposed a curvature-based quality metric. SSIM is a method for predicting the perceived quality of digital images and videos. It compares the structural similarity between a distorted image and a reference image, and derives a quality score for the distorted image. In the method [66], given a reference model and a distorted model, the curvature of each vertex is computed. The mean, standard deviation and covariance of the curvature among the local neighborhood of each vertex are computed. The method computes the difference in mean, standard deviation and covariance of curvature between each pair of vertices in the reference and distorted models. The difference is then pooled into a quality score for the distorted model. Lavoué *et al.* [64] later extended the work [66] so that a multi-scale distance is computed and the cases with changed connectivity of distorted

models can be handled.

Wang *et al.* [136] defined the local roughness as the Laplacian of discrete Gaussian curvature for each vertex on a model, and derived a global roughness measure by computing the integral of the local roughness on the surface. The local roughness is modulated by a visual masking module. The quality of a model is measured through comparison between the global roughness measure of the model and that of the reference model.

The studies [128, 129] employ both curvature amplitude and principal curvature directions for quality evaluation of 3D models. The methods compute the local tensor distance between vertices of the reference and distorted models, and weight the tensor distance by a roughness measure to account for the visual masking effect. The methods provide good results in terms of correlation to the subjective scores in existing databases.

Image-based metrics [62, 71, 72, 147] employ 2D image metrics to evaluate the visual quality of 3D models. The basic idea is to render a number of different views of the 3D models and evaluate the quality of the 3D models by pooling the evaluated visual quality of the 2D images. The major limitation of the image-based metrics is that the evaluation is view-dependent, however, the 3D models are viewed freely by observers with interaction allowed (rotation, translation, scaling). Studies [110, 139] have demonstrated that the perceived quality of still images may not be able to reflect the perceived quality of the corresponding 3D model. Therefore, in this work, we will focus on model-based quality metric for 3D models.

Chapter 3

Perceptual Rendering Based on Visual Saliency

In Subsection 2.2 above, we have introduced existing perceptual rendering algorithms based on visual saliency. The computational cost decreases in saliency-based perceptual rendering by decreasing computational resources distributed to image pixels that are not salient to human observers. Existing schemes use saliency level as the only guide for computation allocation, and do not consider the rendering complexity of pixels. As a result, severe rendering artifacts may be introduced to some non-salient regions, degrading the overall perceived quality of rendered images.

In the study of this chapter, we explore how to incorporate the visual saliency in the rendering process seamlessly to reduce rendering cost. More specifically, we propose a saliency-based perceptual rendering scheme for ray-tracing based rendering algorithms, by adaptively distributing samples among pixels according to the visual saliency and the rendering complexity. In ray-tracing based rendering algo-

rithms, a large number of sample rays are traced for each pixel, and the process is time-consuming. The proposed perceptually based adaptive sampling can reduce the number of samples computed for pixels with low saliency and rendering complexity values, without degrading the perceived quality of rendered images.

We conduct experiments to compare the performance of the proposed visual saliency based perceptual rendering with/without consideration of rendering complexity. Experimental results show that, with the same computational resources, the perceptual rendering scheme which considers both visual saliency and rendering complexity achieves better perceived quality, compared with the perceptual rendering merely based upon visual saliency. To measure the reduction of computational cost by the proposed method, we compare the performance of the proposed method with a recent non-perceptual adaptive sampling scheme. Our experimental results show that the computational cost of the proposed method reduces by about 40%, without loss in perceived quality of rendered images.

3.1 Introduction

Visual saliency based perceptual rendering algorithms [14, 43, 52, 68, 69, 77, 148] have been introduced to decrease computational cost by decreasing computational resources allocated to non-salient pixels that a human viewer pays less attention to. In rendering applications where the rendering process is a passive process, e.g. game and virtual tour, visual saliency information is useful, since the render does not what needs to be rendered in good quality. In the existing methods, the saliency levels of pixels in a coarsely-rendered input image are detected first, usually by the use of a

saliency detection model (such as the one in [47]). In the formal rendering stage, less computational resources are distributed to pixels with lower saliency levels. Existing saliency-based perceptual rendering schemes use saliency level as the only guide for computation allocation. However, rendering process has its own characteristics, and these characteristics have not been considered in the existing saliency-based perceptual rendering algorithms.

In the study of this chapter, we focus on incorporating visual saliency in ray-tracing based rendering algorithms. Ray-tracing based rendering algorithms are used to synthesize high-quality realistic images by accurately simulating light transport [33]. A number of sample rays are traced from each pixel, and the pixel value is computed from the radiance returned by the rays. Increasing the number of samples improves the rendering quality but also increases the rendering cost. To decrease the overall rendering cost, saliency-based perceptual rendering decreases the number of samples computed for non-salient pixels. However, problems will be introduced in the rendering process, if the characteristics of the rendering process are not taken into account in saliency-based perceptual rendering. We elaborate this point with more details and reasoning as follows.

Firstly, rendering complexity varies across pixels in a rendered image. Rendering complexity depends on both the 3D scene and the rendering algorithm. In path tracing and other ray-tracing based algorithms, light rays are traced from image pixels into the 3D scene, and the complexity of each pixel is determined by properties of object surfaces, illumination and the camera. Figure 3.1 gives examples for this, in which two salient regions (human statue and dragon) are shown in Figure 3.1(b) - (e). Figure 3.1(a) is rendered with the standard Monte Carlo path tracing algorithm [54],

Figure 3.1: An example showing that increasing the sampling rate at salient region is not always helpful in quality enhancement. (a) A part of a rendered image of the Statues scene with sampling rate of 1024spp (samples per pixel). (b) and (d) Two salient regions from an image rendered at 16spp. (c) and (e) Two corresponding regions from (a). Note that the increase in number of samples from (b) to (c) does not significantly improve the rendering quality of the human statue part.

which is a widely used rendering algorithm based on ray-tracing. In path tracing, a number of sample rays are computed for each pixel. By comparing the quality of Figure 3.1(b) and (d) with those of (c) and (e), we can find that the sampling rate of 16spp (samples per pixel) is not sufficient for the dragon statue, but is sufficient for the human statue to reach the nearly perfect quality. Although the human statue belongs to a salient region, the rendering complexity for the “matte” material (modeled in PBRT [102]) on the statue is low, so a high sampling rate is unnecessary. In other words, for pixels with low rendering complexity, there is no need to allocate a large amount of computational resources there, even if the saliency values for the pixels are high.

Figure 3.2: An illustration of large distortion turning a non-salient region to a salient one. (a) Part of an image of the Room scene, rendered with high quality at 1024spp. (b) The version with large distortion introduced at the background, rendered at 32spp.

Secondly, under a limited computational budget, rendering distortion may turn a region that should not be of high saliency into a salient region. Research has indicated that large distortion in an image can change the observers’ eye fixations and the associated durations [133], i.e. non-salient regions may be turned to salient ones if a large distortion is introduced. This is illustrated with an example in Figure 3.2, where the background (the back wall) is a non-salient region according to a saliency map (e.g. using the saliency detection model in [34] or [47]) but due to the noise introduced, it attracts human attention as shown in Figure 3.2(b). By simply allocating the computational resources according to the saliency level, the rendering quality of some originally non-salient regions suffers from severe distortion and such regions are turned into salient regions.

3.2 Saliency-based Perceptual Rendering

In order to overcome the two limitations mentioned above, we propose a perceptually based adaptive sampling scheme. In the proposed scheme, samples are adaptively distributed among pixels according to both the visual saliency and the estimated rendering complexity. Fewer number of samples are distributed to pixels with low saliency and rendering complexity, so that the overall computational cost decreases. Existing adaptive sampling schemes [25, 26, 37, 55, 70, 87, 99, 111, 112, 119, 140] have been proposed to decrease the number of samples, by iteratively distributing samples to pixels with high rendering complexity [124]. However, the visual saliency has not been considered in adaptive sampling. In the proposed scheme, we incorporate the visual saliency in adaptive sampling to overcome the limitations in existing perceptual rendering schemes that merely based upon the visual saliency, so that rendering cost can be decreased without perceptual loss in the quality of rendered images.

The block diagram of the proposed scheme is illustrated in Figure 3.3. In the preparation stage of rendering, we sample the image plane uniformly with a small number of initial samples at each pixel. A coarse image is rendered by computing the direct illumination of the sample rays (i.e. one sample ray experiences at most one bounce from the camera to light sources), as a quick estimation of the final image. This coarse image is subsequently used as input to the computation of a saliency map by a saliency detection method. Any existing visual saliency detection method can be incorporated in the proposed scheme, and we use a PFT (Phase of Fourier Transform) based saliency detection method [34] as an example. In the adaptive sampling stage, samples are distributed in an iterative way. Each iteration is composed of four steps:

Figure 3.3: Block diagram of the proposed adaptive sampling scheme.

computing sample values, estimating the complexity map, computing the priority map, and adaptively distributing new samples. In each iteration, a proportion of the sample budget is distributed adaptively to pixels with the highest priority values, until the sample budget exhausts.

By using the proposed perceptual rendering scheme, the above mentioned problems with saliency-based perceptual rendering can be addressed: for pixels with high saliency levels but low rendering complexity, the overall priority value will decrease and the computational resources they receive will also decrease (in comparison with the result using only saliency values); on the other hand, the quality of non-salient regions with high complexity can be enhanced due to the higher priority values given by the priority map and thus non-salient regions with high rendering complexity are less likely to be turned into salient ones due to rendering artifacts.

3.2.1 Saliency Detection with the Phase of Fourier Transform (PFT)

For the saliency detection method in the perceptual rendering scheme, we choose the PFT based saliency detection model [34] as an example. The PFT applies Fourier transform to intensity channel and two color channels in an image:

$$F_i(u, v) = |F_i(u, v)|e^{j \cdot Ph(F_i(u, v))} = \mathfrak{F}(L_i(x)) \quad (3.2.1)$$

where $L_i(x)$ is the pixel value of the Red/Green color channel ($i=1$), Blue/Yellow color channel ($i=2$) or intensity channel ($i=3$) at pixel x ; $\mathfrak{F}$ denotes the Fourier Transform; $F_i(u, v)$ is the corresponding frequency (represented by (u, v)) response. The amplitude $|F_i(u, v)|$ specifies the occurrence rate of each sinusoidal component present in an image, and the phase spectrum $Ph(F_i(u, v))$ specifies where each sinusoidal component locates.

Image locations with less homogeneity have more high-frequency components; by setting the amplitude as a constant (set to one in this work for simplicity) and performing reverse Fourier transform $\mathfrak{F}^{-1}$, we have

$$S(x) = \mathcal{G} * \sum_i ||\mathfrak{F}^{-1} [e^{j \cdot Ph(F_i(u, v))}]||^2 \quad (3.2.2)$$

where $\mathcal{G}$ is a 2D Gaussian filter; $*$ is the convolution operator. With Equation 3.2.2, regions of less homogeneity (which are more likely to be regions with high saliency) will “pop-out”, i.e. $S(x)$ to have large values.

Figure 3.4(b) shows the saliency detection result for the Cornell scene. In the saliency map, higher brightness indicates higher saliency value. As shown in the

Figure 3.4: Saliency map of the Cornell scene: (a) The Cornell scene; (b) Saliency map generated with the PFT.

saliency map, the glass ball and some parts of the two boxes attract more attention (with higher saliency) than the other regions in the rendered image.

3.2.2 Complexity Estimation in Rendering

The computational complexity in rendering is determined by the complexity of the materials/shaders, the number and types of light sources, and the complexity of the rendering equation [54] that is used to compute the global and local illumination [36], and thus it is difficult to measure the computational complexity directly. In adaptive sampling, rendering complexity is estimated in an indirect way, that is, to detect the noise (error) level of different pixels during rendering. With the same sampling rate, pixels displaying higher levels of error usually require a larger amount of computational resources for rendering. Existing adaptive sampling schemes [37, 55, 87, 102] use a sample contrast metric to estimate the error level:

$$\varepsilon'(x) = \frac{1}{V_x} \sum_{i \in \text{Area}(x)} \frac{|s_i - \bar{s}_x|}{\bar{s}_x} \quad (3.2.3)$$

where $\varepsilon'(x)$ is the estimated error value for pixel x ; V_x is the number of samples within the region $Area(x)$ in the pixel; s_i is the value of a sample for this pixel; $\bar{s}_x$ is the average sample value of all the samples for pixel x ; and $|\cdot|$ computes the absolute value.

We use the variance of samples to replace the contrast metric, since the contrast metric always gives high values to dark regions, where $\bar{s}_x$ is small, no matter whether the noise level is high or not. More specifically, the per-pixel noise level is computed by the standard deviation:

$$\varepsilon(x) = \sqrt{\frac{1}{V_x} \sum_{i \in Area(x)} (s_i - \bar{s}_x)^2} \quad (3.2.4)$$

where $\varepsilon(x)$ is the estimated error of pixel x . The calculation with Equation 3.2.4 estimates the error level for a pixel; in general a pixel with high level of error requires more samples for the correct rendering. The estimated error level of each pixel is stored in a complexity map, and higher values indicate that more samples are required.

Figure 3.5(a) illustrates the complexity map computed at the sampling rate of 16spp for the Cornell scene, and higher brightness in the figure indicates higher error value (calculated as $\varepsilon(x)$). The scene is rendered by the path tracing algorithm. In path tracing and other ray-tracing based algorithms, light rays are traced from image pixels into the 3D scene, and the complexity of each pixel is determined by properties of object surfaces, illumination and the camera. In this example, the front face of the cube is not visible to the light source on the top of the box, which means it is illuminated by indirect lighting, i.e. light reflected or refracted from other surfaces in the scene. In path tracing algorithm, a large number of sample rays need to be traced for pixels which are illuminated entirely by indirect lighting. Therefore, the

Figure 3.5: Error values as estimation of rendering complexity: (a) Complexity map computed at the sampling rate of 16spp for the Cornell scene; (b) Image rendered with 16spp; (c) scattering plot of the PSNR versus the average $\varepsilon(x)$ values of pixels in each 8×8 image block in (b).

rendering complexity for the front face of the cube is high.

A image rendered with 16spp is shown in Figure 3.5(b). Figure 3.5(c) presents the PSNR value of each 8×8 image block in Figure 3.5(b) versus its average sample error value ($\bar{\varepsilon}$) for pixels in the block. The PSNR value is obtained by comparing each image block in Figure 3.5(b) with the corresponding block in a reference image shown in Figure 3.4(a) (i.e. the image resulting from a high sampling rate rendering, with 1024spp in this example). Figure 3.5(c) shows that under uniform sampling, rendering quality of image regions decreases as the error value increases; this confirms that error value computed with Equation 3.2.4 can be used as an indicator of the rendering complexity of different image regions.

We first normalize the complexity value $\varepsilon(x)$ to the range of $[0,1]$ and then calculate the priority value $\Theta(x)$ at pixel x as the sum of the saliency value $S(x)$ and the

Figure 3.6: The priority map generated by the proposed method.

complexity value $\varepsilon(x)$, as shown in Equation 3.2.5.

$$\Theta(x) = S(x) + \varepsilon(x) \quad (3.2.5)$$

Equation 3.2.5 indicates that both the complexity and the saliency are important. If the saliency is used as the only guidance in sample allocation, two cases may occur: (i) if the saliency value is high, there will be sufficient rendering, however, computation will be wasted if complexity is low, since low complexity means that a small amount of computation is sufficient for good rendering quality, no matter the pixel is salient or not; (ii) if the saliency value is low, the assigned computation will be low, and noise will be noticeable if rendering complexity is high. Visible noise may turn non-salient regions into salient ones and degrade the overall quality of the rendered image.

A priority map generated at 16spp for the Cornell scene is shown in Figure 3.6. In this figure, higher brightness indicates higher priority value. In the adaptive sampling process, new samples are distributed to pixels with the highest priority values, until the sample budget exhausts.

3.3 Experimental Results and Discussions

In this section, we compare the performance of the proposed saliency-based rendering scheme with the perceptual rendering merely based upon the visual saliency, to demonstrate that the proposed method addresses the limitations mentioned in Section 3.1. Based on the comparison results, we further compare the performance of the proposed rendering scheme with a recent non-perceptual adaptive sampling scheme, to measure the decrease in rendering cost achieved by the proposed saliency-based scheme.

3.3.1 Comparison of Saliency-based Perceptual Rendering

One limitation with perceptual rendering merely based upon visual saliency is that the extra samples distributed to salient pixels with low rendering complexity are wasted. Using the complete Statues scene (part of it has been shown in Figure 3.1), Figure 3.7 shows the close-ups of a salient region (the human statue) which is with high saliency level but low rendering complexity. According to the saliency map in Figure 3.8(a), the human statue region highlighted in Figure 3.7(a) belongs to the salient part. Thus with perceptual rendering merely based upon the saliency map, a high average sampling rate of 241spp (compared with the average sampling rate of 128spp) is assigned to this region. With the proposed perceptual rendering algorithm based on the priority map, however, this region has low rendering complexity (compared with the neighboring regions); therefore a lower sampling rate of 121spp is assigned to this region. As can be seen, the quality of the highlighted region rendered with the saliency map (as shown in Figure 3.7(b)) is similar (and close to that of the reference

Figure 3.7: Comparison of images rendered perceptually with the saliency map and the priority map (I): the bounded region (the human statue) shown in the left picture has high saliency but low rendering complexity. (a) Reference image rendered uniformly with 1024spp and the close-up of a salient region; (b) Image perceptually rendered with the saliency map with an average sampling rate of 128spp and the close-up rendered at an average sampling rate of 241spp; (c) Image perceptually rendered with the priority map at an average sampling rate of 128spp and the close-up rendered at an average sampling rate of 121spp.

shown in Figure 3.7(a)) to the quality of the highlighted region rendered with the priority map (as shown in Figure 3.7(c)). However, the rendering guided by the priority map uses only half of the samples in comparison with the rendering guided by the saliency map. This example demonstrates the advantage of incorporating the rendering complexity in avoiding excessive allocation of samples in the regions with low computational complexity; the saved samples can be used in other regions in need for better overall rendering quality.

Another limitation with perceptual rendering merely based upon visual saliency is that large distortion is introduced in non-salient regions with high rendering

Figure 3.8: Saliency map and priority map for the Statues scene. (a) Saliency map computed using the PFT; (b) Priority map computed with the proposed method.

complexity and turn these originally non-salient regions into salient ones. Figure 3.9 illustrates rendering results of the Sponza scene. Figure 3.9(a) shows a reference image rendered at an average sampling rate of 2048spp. Figure 3.9(b) shows an image rendered at an average sampling rate of 1024spp with the reference method. Figure 3.9(c) shows an image rendered at an average sampling rate of 1024spp, by using the saliency map as the only guide for sample allocation. Figure 3.9(d) shows an image rendered at an average sampling rate of 1024spp, by using the priority map as the guide for sample allocation. The saliency map and the priority map are shown in the top-right corner of Figure 3.9(c) and Figure 3.9(d), respectively. In the maps, higher brightness indicates higher saliency/priority value. Two regions (R1 and R2) are highlighted in the images with rectangles and the close-ups are also provided at the bottom.

The average saliency value (S), complexity value (ε), priority value (Θ) and sampling rate (ρ) of the two regions are listed in Table 3.1. The saliency values of R1 (0.09) and R2 (0.08) are not high, but the complexity values (0.45 and 0.49 for R1 and R2, respectively) are higher, since the two regions are lit by indirect illumination

Figure 3.9: Comparison of images rendered perceptually with the saliency map and the priority map (II): (a) Reference image rendered at an average sampling rate of 2048spp; (b) Image rendered at an average sampling rate of 1024spp with the reference method; (c) Image perceptually rendered with the saliency map at an average sampling rate of 1024spp; (d) Image perceptually rendered with the priority map at an average sampling rate of 1024spp.

reflected from other parts of the scene and indirect illumination usually means high computational complexity. With the perceptual rendering merely based upon visual saliency, a small number of samples (378spp and 367spp on average for R1 and R2) are distributed to the two regions. As a result, visible noise occurs on the two regions, as shown with the close-ups in Figure 3.9(c). With the perceptual rendering guided by the priority map, a larger number of samples (1095spp and 1116spp on average for R1 and R2) are distributed to the two regions, and both the perceived quality (as shown in Figure 3.9(d)) and the objective quality (i.e. PSNR values in Table 3.1) of the two regions are increased.

Table 3.1: Comparison of sampling rates and rendering quality of non-salient regions (R1 and R2) highlighted with rectangles in Figure 3.9.

Region	R1	R2
Average S	0.09	0.08
Average ε	0.45	0.49
Average Θ	0.27	0.28
Average $\rho(\text{spp})$ using saliency map	378	367
Average $\rho(\text{spp})$ using priority map	1095	1116
PSNR(dB) using saliency map	33.2	34.7
PSNR(dB) using priority map	38.4	39.2

Figure 3.10: Comparison of images rendered perceptually with the saliency map and the priority map (III): (a) Reference image rendered at an average sampling rate of 512spp; (b) Image rendered at an average sampling rate of 256spp with the reference method; (c) Image perceptually rendered with the saliency map at an average sampling rate of 256spp; (d) Image perceptually rendered with the priority map at an average sampling rate of 256spp.

Table 3.2: Comparison of sampling rates and rendering quality of non-salient regions for the two regions (R3 and R4) highlighted with rectangles in Figure 3.10.

Region	R3	R4
Average S	0.25	0.03
Average ε	0.95	0.42
Average Θ	0.60	0.22
Average ρ (spp) using saliency map	183	23
Average ρ (spp) using priority map	643	228
PSNR(dB) using saliency map	29.9	27.9
PSNR(dB) using priority map	34.8	37.2

Figure 3.10 illustrates another example for the Killeroos scene. Two regions (R3 and R4) are highlighted in the images, and the average saliency value (S), complexity value (ε), priority value (Θ) and sampling rate (ρ) of the two regions are listed in Table 3.2. The two regions are with low saliency values, but higher rendering complexity, due to the motion blur rendering effect of the killeroos. Low sampling rates are used for the two regions with the perceptual rendering scheme guided by visual saliency only, and artifacts are introduced into the rendered images (refer to Figure 3.10(c) and PSNR values in Table 3.2). The problem can be addressed by the proposed method guided by the priority map: more samples are distributed to the two regions, and the rendering quality of the two regions has improved (refer to Figure 3.10(d) and PSNR values in Table 3.2).

The three examples demonstrate the advantages of the proposed perceptual rendering algorithm in saving rendering cost in regions with high saliency but low complexity, and also in improving rendering quality of regions with high rendering com-

plexity, compared with perceptual rendering based upon the saliency map only.

3.3.2 Cost Reduction Measurement

To measure the reduction of computational cost achieved by the proposed saliency-based adaptive sampling scheme, we compare the proposed method with a recently proposed non-perceptual adaptive sampling method [55]. Cost saved by the proposed perceptual rendering method is estimated through an equal-quality comparison between the method [55] and the proposed method. Reference images are rendered at high sampling densities by the method [55], and test images are rendered at lower sampling densities by the proposed method. In this study, we have conducted a subjective experiment to test whether the perceived quality of reference images and test images are the same.

Six scenes (Cornell, Dragons, Killeroos, Room, Sponza, and Statues) are used in the subjective test. The scene “Cornell”, created at the Cornell University Program of Computer Graphics, is a widely used test scene in rendering. The scenes “Dragons”, “Killeroos” and “Sponza” are from the PBRT [102], and the scenes “Room” and “Statues” are downloaded from Internet. We select these scenes because they all have distinct salient regions, and we can test whether decreasing computation in non-salient regions does not degrade the perceived quality of rendered images. Twenty-four participants were involved in the subjective test. All the participants are students or staff from the School of Computer Engineering, Nanyang Technological University, Singapore. All the participants are with normal vision or corrected to normal vision. Participants are naïve about the experiment purpose. Before the formal test, a short

introduction was given on the test procedure.

During the experiment, six groups of images, corresponding to these six scenes, were presented one by one. Each time, one group of images were shown on the screen: one reference image on the top of the screen and two images on the bottom of the screen. The reference image is rendered by the adaptive sampling method in [55]; one of the bottom images is the same as the reference image, and the other bottom image is rendered by the proposed rendering method, at a lower computational cost. Participants performed the 2AFC (two alternatives forced choice) [5] tasks between the two bottom images, i.e. picking up one image which they thought was more similar to the reference image. The 2AFC (two alternatives forced choice) task is a psychophysical method for eliciting responses from a person about his or her experiences of a stimulus. In the subjective experiment, participants are given two choices, and they have to choose one from the two choices. The 2AFC task has been successfully used in existing studies on perceptual rendering, such as [51, 92, 123], to validate that the images rendered by perceptual rendering methods are visually indistinguishable from the reference images. The order of the six scenes being shown as well as the test image being shown at the left (or right) position was fully randomized.

Table 3.3 shows the experimental results of the subjective viewing test. For each pair, a result of 50% to 50% (12/12) selection is the expected result when no noticeable differences exist between the reference images and the images by the proposed method. As observed from Table 3.3, about a half (47% on average) of the participants thought that images rendered by the proposed method were more similar to the reference images than the reference images themselves. Among all the six

Table 3.3: The numbers of viewers in favor of images rendered by the two algorithms
(24 participants in total)

Scene	Reference	Proposed
Cornell	15	9 (38%)
Dragons	12	12 (50%)
Killeroos	14	10 (42%)
Room	15	9 (38%)
Sponza	10	14 (58%)
Statues	11	13 (54%)

scenes, the highest percentage occurs on the Sponza scene (58%), while the worst cases occur on the Cornell (38%) and Room scene (38%). However, it does not mean that the test image is more similar to the reference image for the Sponza scene than the Cornell and Room scenes. In fact, it is not sufficient to conclude whether the test images are visually indistinguishable from the reference images, based on the results in Table 3.3. Therefore, we have further performed statistical analysis on the results of the subjective test. We conducted the χ^2 (chi square) test to see whether the number of participants who correctly picked up the reference images occurs by chance (if $p > 0.05$) or follows a pattern of preference (if $p < 0.05$). Table 3.4 shows the results of the χ^2 analysis. In this table, χ^2 value resembles a normalized sum of squared deviations between observed and expected frequencies (12/12), and p is the probability of getting a number equal or bigger than the observed one, assuming the null hypothesis (reference images and test images by the proposed method are chosen with equal probability). According to the table, $p > 0.05$ for all the test images by

Table 3.4: Output for a χ^2 test

Scene	
Cornell	$\chi^2 = 1.500, p = 0.221$
Dragons	$\chi^2 = 0.000, p = 1.000$
Killeroos	$\chi^2 = 0.667, p = 0.414$
Room	$\chi^2 = 1.500, p = 0.221$
Sponza	$\chi^2 = 0.667, p = 0.414$
Statues	$\chi^2 = 0.167, p = 0.683$

the proposed method. It indicates that the results shown in Table 3.3 are what we expected; that is, the user preference of images is by chance. It can be concluded from experimental results in Table 3.3 and Table 3.4 that the participants were not able to distinguish the test images rendered by the proposed method from the reference images.

Table 3.5 lists the computational cost (in seconds) for the test images and the reference images. The overall cost of the proposed scheme is the sum of three components: coarse image generation, saliency map computation, and the main rendering process. Note that the complexity map is computed in every iteration of the adaptive sampling process, and the cost of the complexity map is included in the cost of the main rendering process. Cost reduction is calculated by comparing the rendering cost of reference images and the overall cost by the proposed method.

The computational reduction ranges from 15% to 60% (43% on average). The computational saving depends on image content, since the time spent on tracing a sample ray varies at different locations on an image, e.g. tracing a ray which hits the

Table 3.5: Cost (in seconds) for the reference images and test images by the proposed method.

	Ref	Proposed				
Scene	Overall	Overall	Coarse	Saliency	Render	Cost
	Time	Time	Image	Map	Time	Reduction
Cornell	2340.3	1987.2	1.0	0.7	1985.5	15%
Dragons	1075.2	425.7	1.6	0.7	423.4	60%
Killeroos	1344.5	562.8	2.8	0.6	559.4	58%
Room	1240.7	834.2	3.5	0.7	830	33%
Sponza	5497.0	2710.8	13.5	0.8	2696.5	51%
Statues	469.7	267.5	40.5	0.7	226.3	43%

light source after a few rebounds takes longer time than tracing a ray that does not intersect an object. The worst case occurs on the Cornell scene, with the achieved computational reduction at 15%. The reason is that the upper part of the glass ball is not as salient as the lower part of the ball according to the PFT saliency detection model (refer to Figure 3.4). If we decrease the sampling densities at non-salient pixels by a large extent, the quality inconsistency in the glass ball would be easily noticed by a human viewer, since a human viewer perceives the ball as a whole. Therefore, to avoid such quality-inconsistency, we do not decrease substantially the sample budget for the test images, and the achieved reduction of cost by the proposed method is limited for the Cornell scene. The best case occurs on the Dragons scene, with the achieved computational reduction at 60%. The reference image and the test image for the Dragons scene are presented in Figure 3.11. The complex material appearance

Figure 3.11: Reference image and test image for the Dragons scene. (a) Reference image rendered at an average sampling rate of 512spp; (b) Image rendered at an average sampling rate of 256spp with the reference method; (c) Test image perceptually rendered with the priority map at an average sampling rate of 256spp.

and reflection effect in the dragons make it difficult for the human viewer to judge whether the test image is similar or not to the reference image. This observation is consistent with the studies in [105, 106], that complex appearances can hide large appearance change. Therefore, a substantial reduction can be achieved for the Dragons scene. Overall, experimental results show that the proposed perceptual rendering scheme decreases the rendering computation without degrading the perceived quality of resultant images.

3.4 Discussions

One problem with the current work is that the additional requirement for object region detection with saliency modeling in rendering applications has not been considered. The saliency detection model used in the current method gives the rough range of

Figure 3.12: Comparison of saliency detection results using different models: (a) the saliency detection result with the model in [47]; (b) the saliency detection result with the PFT [34].

image regions where human eyes may focus on, but the boundaries of salient region detected may not correlate well with the object boundaries, meaning that for a specific object, some parts of it may belong to the salient regions while some other parts may belong to the non-salient regions.

Figure 3.12(a) and (b) show the saliency maps of the Room-2 scene (Figure 3.13(a)) generated with the saliency detection model in [47] and the model in [34]. In Figure 3.12(a) and (b), some parts of the chair cushion belong to salient regions while some other parts belong to the non-salient regions. However, the large variety of saliency levels within an object will result in an unnatural rendering result, since if the sampling rates vary too fast for an object, the difference in rendering quality can be easily found. In the perceptual rendering results in Figure 3.13(c) and (d), some parts of the salient objects (e.g. the desk) are rendered with better quality, compared with a low-quality image in Figure 3.13(b). However, in Figure 3.13(c) and (d), the artifacts on the chair cushion are obvious, due to the large quality contrast on the

Figure 3.13: Comparison of perceptual rendering results using different saliency detection models: (a) reference image of the Room-2 scene rendered with 1024spp; (b) image of the Room-2 scene rendered with 64spp using the reference method; (c) perceptually rendered image with 64spp using the model in [47]; (d) perceptually rendered image with 64spp using the PFT [34].

cushion region.

To avoid this problem, the whole salient objects, including the accurate boundaries, need to be detected. Accurate segmentation is not trivial in natural images, since the 3D scene information is not available. However, in rendered images from 3D models, the accurate object segmentation information can be easily extracted from the 3D scene. This is a clear advantage of a graphics visual saliency detection method against a natural image saliency detection method, with which accurate object boundaries are not possible to extract in general. To address the above problem, we propose an enhanced visual saliency detection method by making proper use of 3D model information. The details of the enhanced saliency detection method will be described in Chapter 4.

3.5 Concluding Remarks

In this chapter, our work on visual saliency based perceptual rendering is presented. The major contributions of this study are: 1) the in-depth demonstration and analysis of the saliency-based perceptual graphic rendering problem to identify the oversight and drawbacks in the existing approaches; 2) the proposal of a perceptual rendering scheme based on both visual saliency and estimation of rendering complexity.

Unlike the existing work which aim to detect only salient regions for rendering purpose, we argue that both the saliency and the rendering complexity have to be considered simultaneously and formulated appropriately, for the use of the available computational budget to achieve the best perceived rendering image quality.

The proposed scheme uses the PFT for the saliency calculation, and employs a dedicated metric, the sample error metric, for estimation of rendering complexity. In the rendering process of the proposed method, samples are iteratively distributed to pixels with high priority value, which is the combination of saliency and complexity values. We have tested and compared the perceptual rendering performance via experiments. The results have firstly confirmed that the proposed scheme overcomes the shortcomings in the existing saliency-based perceptual rendering methods with the cases of both low and high rendering complexity. The subsequent subjective experiments have shown that, the proposed scheme decreases the rendering computation without degrading the perceived quality of rendered images, compared with a non-perceptual adaptive sampling scheme. The proposed scheme in this chapter will be used as a base-line rendering scheme in Chapters 4 and 5.

We have also discussed the possibility to improve the saliency detection, and the

details of the enhanced saliency detection scheme will be described in Chapter 4.

Chapter 4

Enhanced Visual Saliency

Detection for Rendered Graphics

It has been mentioned in Chapter 3 that it is difficult to detect the accurate boundaries of salient objects for saliency detection methods in natural images, due to the lack of 3D scene information. In this chapter, we propose an enhanced visual saliency detection method to detect salient objects in rendered graphics. Different from existing graphics visual saliency detection methods, which estimate saliency based on pixel-level contrast, the proposed saliency detection method detects salient objects by computing object-level contrast. Given a rendered image, the proposed method first extracts dominant colors from each object, and represents each object with a dominant color descriptor (DCD). Saliency of each object is then calculated by measuring the contrast between the DCD of the object and the DCDs of its surrounding objects. We also design a new iterative suppression operator to enhance the saliency result.

Compared with existing visual saliency detection methods, the proposed method can obtain much better performance in salient object detection. We further apply the enhanced visual saliency detection method to saliency based perceptual rendering. Compared with the relevant existing saliency detection methods, the enhanced visual saliency detection method yields more consistent rendering quality for each salient object.

4.1 Introduction

Many studies show that objects are units of visual attention and information processing, and the saliency-driven attention mechanism is object-based rather than location-based [49, 94, 114, 143]. The object-based saliency mechanism explains why the change in quality on a salient object in a rendered image can be easily detected by a human viewer and degrades the perceived quality of the rendered image. Besides the application in perceptual rendering [68, 77, 144], saliency detection methods have also been exploited in various computer graphics applications, such as generation of various visual effects [43], and exploration in VE (virtual environment) [43, 68]. Considering the wide applications of saliency detection methods in 3D graphics, it is important to design a saliency detection method to detect the accurate boundaries of salient objects.

In the area of computer vision and image/video processing, many computational methods have been devised to detect salient regions for various visual signal processing applications, such as object detection [95], object tracking [98], and image retargeting [28, 35]. Besides the applications in natural images/videos processing,

saliency detection methods [43, 68, 77, 144] have also been proposed in the context of 3D graphics. To detect salient regions in rendered images, graphics saliency detection schemes exploit both image features and model features extracted from 3D scenes, such as pixel depth and object size. The graphics saliency detection methods in [43, 68, 77] are all designed based on the framework of a saliency detection method for natural images [47]. Existing graphics saliency detection methods choose the spatial contrast based saliency detection method [47] rather than the frequency domain analysis based methods, such as [34, 45], as the basis, since the extracted information from 3D models (e.g. depth and object index) are in the spatial domain, and can be easily integrated with image features in the computation of contrast in the spatial domain. In the graphics saliency detection methods [43, 68, 77], first, features are extracted from rendered images as well as 3D models. Then saliency is computed as pixel-level contrast between features of the center pixels and the neighboring pixels: pixels with larger contrast to their surroundings are assigned higher saliency values.

Longhurst *et al.* [77] proposed a visual saliency detection method for animations (a sequence of rendered images). In the method, they made use of both image features, such as pixel color and intensity, and 3D model features including motion, depth, and habituation (which accounts for how long an object has been present in an animation). It is obvious that the saliency detection model based on pixel-level contrast cannot detect entire salient objects.

Lee *et al.* [68] proposed a method to detect visually attended objects in VE. In the method, the top-down mechanism, which is related to prior knowledge of observers, is also taken into account besides the bottom-up mechanism. To detect salient objects, they first detect salient pixels by computing pixel-level contrast and then average the

saliency values of pixels belonging to the same object. To distinguish different objects in a rendered image, the object index information is extracted from 3D models. An index number is a unique number given to each object in a rendered image, and it is shared by image pixels belonging to the same visible object. The problem of the scheme is that it cannot detect salient objects which include some low-contrast regions.

Recently, Hillaire *et al.* [43] proposed a method to detect visual fixation in VE by considering both top-down and bottom-up mechanisms. In the bottom-up saliency detection scheme, pixel depth information is used together with image-domain features for saliency detection. It is noted that the scheme aims to detect gaze positions where observers' eyes fixate on, and it is not able to detect the whole salient objects, as shown in Figure 4.1(c).

The common problem with existing saliency detection methods is that they are not able to detect the boundaries of salient objects accurately. In pixel-level contrast computation, the object boundary information available in 3D models is not used, and some pixels of salient objects may not be detected. Figure 4.1 shows examples of the problem. Figure 4.1(a) is a rendered image of the Living room-1 scene; Figure 4.1(b) shows the human labeled ground-truth map (white parts are salient regions and black parts are non-salient regions). Figure 4.1(c) - (e) shows the saliency detection results from the methods in [43], [68], and the proposed method, respectively. In these maps, a higher gray-level value indicates a higher level of saliency. Compared with the ground-truth map in Figure 4.1(b), some pixels in the highlighted painting are assigned low saliency values in Figure 4.1(c), due to the low contrast of these pixels to their neighborhood. In the saliency map shown in Figure 4.1(d), the highlighted

Figure 4.1: The problem of existing visual saliency detection schemes. (a) Rendered image of the Living room-1 scene (the region highlighted with a rectangle corresponds to the highlighted regions in (c) and (d)). (b) The human-labeled ground-truth map. (c) Saliency map by the scheme in [43]. (d) Saliency map by the scheme in [68]. (e) Saliency map by the proposed scheme.

painting is not detected. Since only parts of the painting (e.g. the black background) are with high contrast, a relatively low overall saliency value is calculated after the averaging operation.

To overcome the problem in existing methods, we propose an enhanced visual saliency detection scheme in this work. The underlying idea is to compute the saliency by measuring object-level contrast instead of computing the pixel-level contrast as in existing work, as well as making good use of the object shape information available from 3D models. Since different objects in rendered images can be distinguished by using 3D scene information, the contrast of each object to the other objects can be directly measured. This is a clear advantage of a graphics visual saliency detection method against a natural image saliency detection method, with which accurate object boundaries are not possible to extract in general. In the proposed method, color features are first extracted from each object and the features are represented by the dominant color descriptor (DCD). Then object-level contrast is calculated as the d-

ifference between the DCD of each object and the DCDs of the surrounding objects. After applying a new suppression operator on the contrast computation result, the final saliency map is obtained.

The contribution of the work in this chapter includes four aspects. First, we introduce a saliency measurement in terms of object-level contrast, which yields detection of boundaries of salient objects in rendered images. Second, we adopt the DCD to represent the features of each object, so that both color information and characteristics of color distribution are considered in computing object contrast. Third, we design an iterative suppression operator, which promotes objects with high contrast, while suppressing objects with low contrast. After the suppression, a saliency map highlighting the most salient objects can be generated. Last, we incorporate the enhanced saliency detection method in perceptual rendering, and demonstrate that the problem of quality change in a salient object, which leads to the decrease of the overall quality of a rendered image, is addressed.

4.2 Enhanced Visual Saliency Concept and Detection

The framework of the enhanced visual saliency detection method is shown in Figure 4.2. Starting from a 3D scene, a coarse image is first rendered, and binary object masks are extracted from the 3D models to distinguish different objects ($O_1, O_2, \dots, O_N$) in the rendered image. Each mask corresponds to one object in the rendered image. Then dominant colors of each object are extracted, and each object is rep-

Figure 4.2: The framework of the proposed visual saliency detection method.

resented by the DCD (dominant color descriptor). In the DCD, the red, green, and blue color channels of the dominant colors are converted to three types of feature channels: red-green color opponency ($\mathcal{F}_{RG}$), yellow-blue color opponency ($\mathcal{F}_{BY}$), and intensity ($\mathcal{F}_I$). After feature extraction, we compute the contrast of DCDs between each object and its surrounding objects, and integrate the contrast values of all the objects into a conspicuity map. The final saliency result is obtained by suppressing the conspicuity map.

4.2.1 Dominant Color Extraction

In the implementation, images are rendered with the Blender, which is an open source 3D computer graphics software supporting fast rendering with scanline ray tracing. In 3D modeling, an object index can be assigned to each object in a scene. Blender is able to generate a binary mask for each object in a rendered image. The first row in Figure 4.2 illustrates examples of a rendered image and two binary object masks corresponding to the image. Given an object index, pixels covered by the object are with the value “1” in the binary object mask, and the other pixels are with the value “0” in the binary object mask. With the binary object masks, each object can be segmented from the rendered image. In the proposed scheme, the DCD is adopted to represent the features of each object. The DCD is one of the color descriptors in MPEG-7, which provides a compact and intuitive description of the representative colors in a region.

Given a region $\mathbf{R}$, the feature descriptor consists of the representative color (c_i), the percentage of each representative color (Ω_i), the variances value of each dominant color (var_i), and the spatial coherency of dominant colors (ρ_s) [84]:

$$f(\mathbf{R}) = \{(c_i, \Omega_i, var_i), \rho_s\}, \quad i = 1, \dots, m \quad (4.2.1)$$

where m is the number of the dominant colors in region $\mathbf{R}$ and $\sum_{i=1}^m \Omega_i = 1$. In this study, the optional parameters var_i and ρ_s are ignored for computational simplicity, and the DCD is described in the color information (c_i) and the color distribution characteristics (Ω_i) of each object.

In this work, K-means algorithm is used in color clustering. Consider an object O_N covering z pixels $(x_1, x_2, \dots, x_z)$, where each pixel is a 3-dimensional vector of red,

green, and blue color channels, K-means clustering aims to partition the pixels into m clusters ($m \leq z$) $O_N = \{O_{N,1}, O_{N,2}, \dots, O_{N,m}\}$ so as to minimize the within-cluster sum of squares:

$$\arg \min_{O_N} \sum_{i=1}^m \sum_{x_l \in O_{N,i}} \|x_l - c_i\|^2 \quad (4.2.2)$$

where c_i is the mean of pixels in cluster $O_{N,i}$. Typically, 3-4 colors provide a good characterization of the region colors [84]. The percentage of each cluster, Ω_i , in object O_N is calculated as the ratio of pixel numbers in cluster $O_{N,i}$ to the overall pixel numbers in the object:

$$\Omega_i = \frac{|O_{N,i}|}{z} \quad (4.2.3)$$

where the $|\cdot|$ operator denotes the number of pixels in the cluster.

At this point, each object O_N can be represented by the DCD: $f(O_N) = \{c_i, \Omega_i\}$, where $i = 1, 2, \dots, m$, in which m is the number of dominant colors in O_N ; $c_i = [r_i, g_i, b_i]$ is the dominant color in RGB color space; and Ω_i is the percentage of the dominant color.

Figure 4.3 shows dominant color results of two objects. In Figure 4.3 (b) and (c), each object is represented with three dominant colors, i.e. the original pixel colors are replaced with the nearest dominant colors. By comparing the objects shown in Figure 4.3 (b) and (c) with the corresponding objects shown in (a), we can find that the major colors of the two objects are well preserved and unimportant details are ignored.

Figure 4.3: Example of dominant color extraction results. (a) A rendered image of the Sponza scene. (b) and (c) Two objects represented with their dominant colors.

4.2.2 Feature Channel Extraction for Each Object

From the red, green and blue color channels of the dominant colors, three types of features can be extracted. Given the red (r), green (g), and blue (b) channels of the dominant color, four broadly tuned color channels are first created [48]:

$$R = r - (g + b)/2 \quad (4.2.4)$$

$$G = g - (r - b)/2 \quad (4.2.5)$$

$$B = b - (r + g)/2 \quad (4.2.6)$$

$$Y = (r + g)/2 - |r - g|/2 - b \quad (4.2.7)$$

where, R , G , B and Y are the red, green, blue and yellow channels (negative values are set to zero), respectively.

Then the two chromatic feature channels, red/green ($\mathcal{F}_{RG}$) and blue/yellow ($\mathcal{F}_{BY}$), which account for responses of the “double opponency” cells in the primate visual cortex [27], are computed as:

$$\mathcal{F}_{RG} = R - G \quad (4.2.8)$$

$$\mathcal{F}_{BY} = B - Y \quad (4.2.9)$$

The intensity feature channel ($\mathcal{F}_I$) is computed as follows:

$$\mathcal{F}_I = (r + g + b)/3 \quad (4.2.10)$$

After the feature extraction, each dominant color of an object can be represented as a 3-dimensional feature vector $c_i = [\mathcal{F}_{RG.i}, \mathcal{F}_{BY.i}, \mathcal{F}_{I.i}]$.

4.2.3 Computation of Object-level Contrast

The saliency of each object is computed as the contrast between the DCDs of the object and its surrounding objects. A Gaussian model of the Euclidean distances between objects is used to determine the weighting for the contrast to surrounding objects; surrounding objects which are nearer to the center object are given larger weights. Consider that there are V visible objects in a rendered image, for object O_N , the global contrast $C(O_N)$ is computed as the weighted sum of the DCD difference of the object to each of the remaining $V - 1$ objects:

$$C(O_N) = \sum_{M \neq N}^{V-1} \frac{1}{\beta \sqrt{2\pi}} e^{-\text{Euc}(N,M)/2\beta^2} \cdot \text{Diss}^2(f(O_N), f(O_M)) \quad (4.2.11)$$

where β is a parameter for the Gaussian model; $\text{Euc}(N, M)$ is the Euclidean distance between the centroids of objects O_N and O_M . In this work, we set β to 4% of the input

image width. If the selected β is small, the fast decay rate in Equation 4.2.11 makes the computed contrast a local one, i.e. the contrast from objects with large distances to the center object is neglected. If the selected β is big, objects with large distances to the center object would be assigned similar weights as objects which are nearer to the center object, which is not desired. $Diss^2(f(O_N), f(O_M))$ is the difference between the DCDs of O_N and O_M , and it is computed with the dissimilarity metric defined by the DCD. Consider the DCDs of two objects,

$$f(O_1) = \{c_{1i}, \Omega_{1i}\}, \quad i = 1, 2, \dots, m_1 \quad (4.2.12)$$

$$f(O_2) = \{e_{2j}, \Omega_{2j}\}, \quad j = 1, 2, \dots, m_2 \quad (4.2.13)$$

where the subscripts 1 and 2 in all variables stand for objects O_1 and O_2 respectively. c_{1i} and e_{2j} are dominant colors represented as vectors of features: $c_{1i} = [\mathcal{F}_{RG-1i}, \mathcal{F}_{BY-1i}, \mathcal{F}_{L-1i}]$ and $e_{2j} = [\mathcal{F}_{RG-2j}, \mathcal{F}_{BY-2j}, \mathcal{F}_{L-2j}]$. Ω_{1i} is the percentage of the i^{th} cluster in object O_1 , and Ω_{2j} is the percentage of the j^{th} cluster in object O_2 . m_1 and m_2 are the numbers of dominant colors in the two objects. The dissimilarity $Diss^2(f(O_1), f(O_2))$ between two DCDs can be computed as [84]:

$$Diss^2(f(O_1), f(O_2)) = \sum_{i=1}^{m_1} \Omega_{1i}^2 + \sum_{j=1}^{m_2} \Omega_{2j}^2 - \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} 2a_{1i,2j} \Omega_{1i} \Omega_{2j} \quad (4.2.14)$$

where $a_{1i,2j}$ is the similarity coefficient between two colors c_{1i} and e_{2j} :

$$a_{1i,2j} = \begin{cases} 1 - Euc_{1i,2j}/Euc_{\max}, & Euc_{1i,2j} \leq T_c \\ 0, & Euc_{1i,2j} > T_c \end{cases} \quad (4.2.15)$$

where $Euc_{1i,2j} = \|c_{1i} - e_{2j}\|$ is the Euclidean distance between two feature vectors c_{1i} and e_{2j} , T_c is the maximum distance for two colors to be considered similar, and

Figure 4.4: Saliency detection results with different T_c and ξ values. (a) A rendered image of the Tribal scene. (b) Saliency detection result at $T_c = 10$, and $\xi = 1.0$. (c) Saliency detection result at $T_c = 20$, and $\xi = 1.5$.

$Euc_{\max} = \xi \cdot T_c$. A recommended value for T_c is between 10-20 and for ξ is between 1.0-1.5 in MPEG-7 [84]. In this study, we select $T_c = 20$ and $\xi = 1.2$, and using other T_c and ξ values within the above mentioned ranges yield similar saliency detection results. Figure 4.4 illustrates the saliency detection results of the Tribal scene, with different T_c and ξ values, and the detection results in Figure 4.4(b) and (c) are similar to each other. The above dissimilarity measure is equivalent to the quadratic distance measure that is commonly used in comparing two color histogram descriptors [21, 84]:

$$Diss_h^2(H_1, H_2) = (H_1 - H_2)^t A (H_1 - H_2) \quad (4.2.16)$$

where H_1 and H_2 are color histogram vectors of length m_h , and the coefficients of matrix A are $a_{1i,2j}$. By computing object-level contrast with Equations 4.2.14 and 4.2.15, both color information and characteristics of color distribution of objects are considered in saliency detection.

4.2.4 Iterative Suppression

After the contrast computation, the next step is to generate a conspicuity map. The conspicuity value of each object corresponds to the computed contrast value of the object with Equation 4.2.11. A suppression operator is then applied on the conspicuity map to reduce the number of salient objects. It is indicated in [67], that without the suppression operation, too many regions will be detected as salient; and the suppression helps to define what makes something unique, and therefore potentially salient. In the proposed method, we design an iterative suppression operator. Assume there are V objects in a rendered image, the suppression $\delta(O_N)$ on object O_N is computed as the average conspicuity value of the other $V - 1$ objects:

$$\delta(O_N) = \frac{1}{V-1} \sum_{M \neq N} C(O_M) \quad (4.2.17)$$

As indicated in Equation 4.2.17, if the conspicuity value of an object is equal or smaller than the average conspicuity value of the other objects in the scene, the conspicuity value of the object is suppressed to zero. Then the normalization operator is as follows:

$$C(O_N) \leftarrow \max(C(O_N) - \delta(O_N), 0) \quad (4.2.18)$$

In every iteration, the conspicuity value of each object is updated with Equation 4.2.18. As the number of iterations increases, higher conspicuity values are promoted and lower values are suppressed. For each object, the final saliency value is the conspicuity value updated after the last iteration. The iteration number is a parameter depending on user requirement; if a more compact saliency map is desired (saliency values concentrate on a few objects), a larger iteration number can be chosen.

Figure 4.5: Saliency maps from different operators. Maps in the first row are generated by the operator in [47]. Maps in the second row are generated by the proposed suppression operator.

Figure 4.5 provides the comparison between saliency maps obtained from one conspicuity map using different operators. The upper row shows saliency maps from an existing normalization operator in [47]. The operator can iteratively yield surround-inhibition on conspicuity values of pixels. The lower row shows saliency maps from the proposed suppression operator. In the saliency maps generated using the operator in [47], most pixels covered by the painting are assigned very low values. In the scheme [47], conspicuity values of pixels are suppressed by their surrounding pixels, i.e. pixels covered by the painting are also subject to suppression from the neighboring pixels on the same object. As a result, conspicuity values of most pixels on the painting are suppressed to zero. The proposed suppression operator prevents self-inhibition of pixels within the underlying object, and hence the conspicuity value of the painting is preserved after the suppression. In addition, as the iteration number increases, saliency values become more concentrated on the most salient objects, as

Figure 4.6: Illustration of the multi-scale contrast computation.

shown in Figure 4.5. The saliency maps shown in the experimental results in Section 4.3 are generated by updating conspicuity maps once with the proposed operator. Of course, a higher iteration number can be chosen if a more compact saliency map is desired; for example, if we hope to detect a single object with the highest saliency in the image, we could choose a high iteration number.

4.2.5 Discussion on Object-level Contrast Computation

4.2.5.1 Comparison between Object-level Contrast and Multi-scale Pixel-level Contrast

In this section, we compare the proposed object-level contrast computation scheme with the multi-scale pixel-level contrast computation scheme used in existing graphics visual saliency detection methods [43, 68, 77]. In multi-scale contrast computation scheme, saliency is calculated based on the contrast between different levels of multi-scale image pyramids, as illustrated with Figure 4.6. The input image is progressively low-pass filtered and subsampled with Gaussian pyramids, and from each finer (lower) level to a coarser (higher) level, the spatial resolution decreases by a factor of 2. Finer-scale map accounts for the “center” (c) while coarser-scale map accounts for

the “surround” (s).

With the multi-scale center-surround difference computation, salient structure of different scales can be detected in the multi-scale feature maps. In a lower level feature map, fine or small-size salient details are detected; in a higher level feature map, larger-size salient regions can be detected. Feature maps of each feature type are then combined into a conspicuity map, through across-scale addition.

Although the multi-scale contrast schemes can improve the performance over single-scale contrast methods, the multi-scale contrast scheme is still not able to detect whole salient objects. When a salient object includes regions which are similar to the surroundings, these regions cannot be detected by existing multi-scale methods. With the proposed scheme based on object-level contrast, however, accurate boundaries of salient objects can be detected.

Figure 4.7 shows the saliency maps of the Living room-2 image. Figure 4.7(b) shows the human labeled ground-truth map (white parts are salient regions and black parts are non-salient regions). Figure 4.7(c) - (e) show the saliency detection results from the methods in [43], [68], and the proposed method, respectively. All the saliency maps shown in Figure 4.7(c) - (e) are generated with the color and intensity features only, to exclude the effect on saliency detection results from using different features.

For each map, we show the false-colored close-ups of two salient objects, so that the saliency distribution in these objects can be better analyzed. In the close-ups, the saliency value increases from blue to red, as shown by the bar on the right of Figure 4.7. We can see that some pixels in the salient objects are assigned low saliency values in the saliency map shown in Figure 4.7(c), which is not in-line with the ground-truth map in Figure 4.7(b). These pixels are detected as non-salient due to the low

Figure 4.7: Comparison of saliency maps computed from multi-scale contrast and object-level contrast. (a) A rendered image of the Living room-2 scene (the two regions highlighted with rectangles correspond to the highlighted regions in saliency maps shown in (c), (d) and (e)). (b) The human-labeled ground-truth map. (c) Saliency map by the method in [43] and close-ups of two false-colored objects. (d) Saliency map by the method in [68] and close-ups of two false-colored objects. (e) Saliency map by the proposed method and close-ups of two false-colored objects.

contrast to the surrounding pixels. For the saliency map shown in Figure 4.7(d), since some pixels of the salient objects are assigned low saliency values by the multi-scale contrast computation, the entire salient objects are assigned medium-level saliency values after the average operation. In contrast, the proposed scheme based on object-level contrast can detect most salient objects (see Figure 4.7(e)).

4.2.5.2 Comparison between Global Contrast and Local Contrast

In the proposed saliency detection scheme, the global contrast of each object to all the other objects is computed. Different from a local contrast based saliency detection scheme which only considers the difference of an object with respect to its local

Figure 4.8: Comparison between saliency maps computed based on local object contrast and global object contrast. (a) A rendered image of the Kitchen scene (the ground-truth map is shown in the bottom right corner). (b) Saliency map based on local object contrast. (c) Saliency map based on global object contrast.

background, the global contrast based scheme considers the disparity of an object in the entire image. Compared with the local scheme, the global contrast based scheme is able to assign comparable saliency values to similar objects in the image.

Figure 4.8 gives a comparison between saliency maps based on local and global object contrast. The two cupboards highlighted with white rectangles in Figure 4.8(a) do not have obvious difference from other cupboards in this scene, and are not labeled as salient in the ground-truth map (the bottom right corner of Figure 4.8(a)). The saliency map shown in Figure 4.8(b) is computed by measuring the local object contrast, i.e. the contrast between the object and its local background. A way of

choosing the local background is to enlarge the object by a scale factor of 2 in both the horizontal and vertical directions, and the local background of the two cupboards are also highlighted in Figure 4.8(a). In the saliency map shown in Figure 4.8(b), the two cupboards are assigned high saliency values due to the large local contrast, which is not in-line with the ground-truth map in Figure 4.8(a). The problem can be solved with the saliency map based on global contrast of objects (see Figure 4.8(b)). Since features of the two cupboards are compared with all the objects in the scene, comparable saliency values are assigned to the two similar cupboards in the scene.

4.3 Experimental Results and Discussions

In this section, we evaluate the performance of the proposed saliency detection algorithm based on the comparison experiments between the proposed method with other existing ones. Our test data include eight rendered images, with the size of 400×400 , and their corresponding ground-truth maps shown in Figure 4.9(a) and (g) respectively. The scenes cover various situations, including indoor scenes and more complex outdoor scenes.

The ground-truth saliency maps are produced by nine participants with Adobe Photoshop. All participants were naïve to the purpose of the study. Before the formal experiment, the participants were given an instruction on how to use the Lasso and Pen selection tools in Photoshop to label image regions. After that, the participants were presented with an example image on a 19-inch LCD monitor with display resolution at 1440×900 pixels. The participants were asked to label the salient regions which attract their attention in the example image, with the Lasso and Pen

Figure 4.9: Saliency maps from different saliency detection methods. (a) Original images. (b) Saliency maps from the method in [47]. (c) Saliency maps from the method in [34]. (d) Saliency maps from the method in [68]. (e) Saliency maps from the method in [43]. (f) Saliency maps from the proposed method. (g) Human-labeled ground-truth maps from nine subjects.

tools. In the formal experiment, the participants were presented with eight rendered images on the monitor, and they were asked to label the salient parts in the images, with the selection tools. The final ground-truth maps, marked as white regions, are obtained by extracting the overlapping salient regions by the nine participants.

In Figure 4.9, the saliency maps from different saliency detection methods are shown, including two saliency detection method on natural images [34, 47], two saliency detection methods for computer synthesized images [43, 68], and the proposed detection method. The PFT saliency detection method in [34] relies on the phase information of Fourier Transform, and it has been used as an example in the saliency-based perceptual rendering scheme, which is described in Chapter 3. We implement the saliency detection components of the work in [68] and [43], while the results by the method in [34, 47] are obtained by the original code published online by the authors.

Figure 4.9(b) are saliency maps generated by the method of [47]. By comparing these saliency maps with the ground-truth maps in Figure 4.9(g), it is observed that, without the accurate object boundary information, some parts of the salient objects cannot be detected. Figure 4.9(c) illustrates saliency maps generated by the PFT method in [34]. In the saliency maps, parts of the edges of salient objects are detected, but some regions inside the objects are not detected. Saliency maps in Figure 4.9(d) are produced by the saliency detection method of [68]. It is shown that some salient objects are not detected. Saliency maps in Figure 4.9(e) are resulted from the method devised by Hillaire *et al.* [43]. It is seen that only pixels with high contrast to their neighborhood are detected. Saliency maps from Figure 4.9(f) are computed with the proposed saliency detection method. We can see that most of the salient objects in

the test images are detected.

To further analyze the results, we give a quantitative measurement to demonstrate the performance. To measure the level of agreement of the saliency maps, three indices are proposed: precision, recall, and F-measure. Precision is defined as the ratio of correctly detected salient pixels to overall salient pixels detected by the saliency detection method. Recall is calculated as the ratio of correctly detected salient pixels by the saliency detection method to the overall target pixels in the ground-truth map. Since precision and recall scores are usually not discussed in isolation, we also compute a single measure, i.e. F-measure, which is a harmonic mean of precision and recall. Given a ground-truth saliency map $\mathbf{G}$ and the binary saliency map $\mathbf{S}$ for an image, the precision, recall and F-measure are calculated as follows:

$$precision = \frac{|\mathbf{G} \cap \mathbf{S}|}{|\mathbf{S}|} \quad (4.3.1)$$

$$recall = \frac{|\mathbf{G} \cap \mathbf{S}|}{|\mathbf{G}|} \quad (4.3.2)$$

$$F\text{-measure} = \frac{(1 + \omega) \cdot precision \cdot recall}{\omega \cdot precision + recall} \quad (4.3.3)$$

where the $|\cdot|$ operator denotes the number of pixels in the area. We set $\omega = 0.3$ as in [32] for more consideration of the precision value over the recall value in this experiment. Before obtaining the three indices, Otsu's thresholding algorithm [97] is used to binarize the saliency map into salient part and non-salient part (1 for salient pixels and 0 for non-salient pixels).

Figure 4.10 shows average results of the eight rendered images in Figure 4.9. It is shown that the method of [47] yields high precision and low recall. That means most pixels detected by the method belong to the salient objects labeled in the ground-truth

Figure 4.10: Experiment results for the comparison between the proposed and other methods: Itti (the method in [47]), PFT (the method in [34]), Lee (the method in [68]), Hillaire (the method in [43]).

maps, but a large number of salient pixels are still not detected. The method of [34] yields high recall and low precision. The detected salient regions by the method of [34] are mainly edges, but edges are not necessarily salient regions, thus the precision of the detection results is low. The methods of [43, 68] generate high recall and low precision scores. It cannot be concluded that most salient objects are correctly detected from the high recall scores only. The low precision scores indicate that a large number of non-salient objects are labeled as salient. In an extreme case, marking all objects in an image as salient gives a highest recall (i.e. 1.0), but all objects are salient means no object is unique or really salient. In contrast, the proposed method yields much

Table 4.1: Cost (in seconds) of the saliency detection methods.

Scene	Itti[47]	PFT[34]	Lee[68]	Hillaire[43]	Proposed
Fairy	1.1	0.7	3.9	1.4	6.3
Gazebo	1.0	0.8	2.5	1.5	4.9
Kitchen	1.0	0.7	2.4	1.5	10.4
Lotus	1.1	0.6	1.8	1.5	5.1
Living room-1	0.9	0.7	1.6	1.5	3.7
Living room-2	0.8	0.7	1.6	1.6	3.5
Sponza	1.0	0.7	2.0	1.5	10.7
Tribal	0.9	0.8	1.7	1.5	4.2

better overall performance with the highest precision, recall and F-measure scores, i.e. most salient objects are detected and the false detection rate is low.

We also compare the cost of the saliency detection methods and the results are shown in Table 4.1. Among the cost of the test methods, the cost of the proposed saliency detection method is the highest. The reason is that, the DCD of each object is compared with the DCDs of all the other objects in the scene, and this process is time-consuming. In off-line rendering, the rendering process may take several minutes, and the cost of saliency computation is much lower than that of the rendering process. However, in real-time rendering, the rendering process takes only several milliseconds, and the cost of saliency computation is too high. In the future work, we are to decrease the cost of the proposed saliency detection method by optimizing the pipeline of the saliency computation.

4.4 Application in Perceptual Rendering

In this section, we incorporate the enhanced visual saliency detection method in the application of perceptual rendering to demonstrate the advantages of the proposed method. We use the saliency-based adaptive sampling scheme proposed in Chapter 3. In the proposed adaptive sampling scheme, a saliency detection method is used for predicting the saliency levels of regions in an image to be rendered. The rendering complexity of pixels are estimated by a sample error metric in Equation 3.2.4. A priority value is computed for each pixel by integrating the saliency value and complexity value. In the adaptive sampling, samples are iteratively distributed to pixels with the highest priority values, until the sample budget exhausts. Here, we compare the performance of the proposed saliency detection method with the relevant existing saliency detection methods in the same adaptive sampling scheme.

Two example scenes, Tribal and Fairy, are shown in Figure 4.11 and Figure 4.12. The scenes are different from the scenes used in the part of work described in Chapter 3 due to the format reason. Some scenes (Cornell, Dragons, Killeroos, Room, and Statues) in Chapter 3 are in “.pbrt” format, which is defined by the authors of PBRT [102] and it is not trivial to convert the format to the widely used formats of 3D scenes such as “.obj” and “.3ds”. Currently, we use the Blender software to extract the object segmentation information from 3D scenes, but the scene files in “.pbrt” format cannot be used as input to the Blender software. Since the object segmentation information of the scenes (Cornell, Dragons, Killeroos, Room, and Statues) cannot be extracted at the current stage, we use different scenes in this Chapter.

Figure 4.11 illustrates the perceptually rendered images for the Tribal scene by

Figure 4.11: Perceptually rendered images for the Tribal scene by using different saliency detection methods. (a) and (b) show reference image rendered uniformly at 1500spp (samples per pixel) and close-up of a highlighted salient region, respectively. (c), (d), (e), (f), (g) show close-up of a corresponding salient region in an image perceptually rendered with an average of 800spp by using the saliency detection method in [47], [34], [68], [43], and the proposed saliency detection method, respectively.

using different saliency detection methods. Figure 4.11(a) and (b) present a reference image rendered uniformly at 1500spp (samples per pixel) and close-up of a salient region highlighted with a rectangle. Close-ups of the corresponding salient regions in images perceptually rendered by using different saliency detection methods are presented in Figure 4.11(c) - (g). According to the saliency detection results in Figure 4.9, most pixels of the tent (except the edges) are not detected by the methods of [34, 47], thus low sampling rates are used for pixels in the tent in Figure 4.11(c) and (d). Visible noise is introduced to the salient object in Figure 4.11(c) and (d),

Figure 4.12: Perceptually rendered images for the Fairy scene by using different saliency detection methods. (a) and (b) shows reference image rendered uniformly at 1500spp (samples per pixel) and close-up of a highlighted salient region, respectively. (c), (d), (e), (f), (g) shows close-up of a corresponding salient region in an image perceptually rendered with an average of 700spp by using the saliency detection method in [47], [34], [68], [43], and the proposed saliency detection method, respectively.

and the computed PSNR values of the tent (36.9dB and 37.5dB) are relatively low. Since the human viewer perceives the tent as a whole, the noise occurring in the area covering the object of the tent is noticeable and degrades the perceived quality of the perceptually rendered image. According to the saliency detection results in Figure 4.9, the major parts of the tent are detected by the methods of [43, 68]. As a result, the rendering quality of the tent in Figure 4.11(e) and (f) is better than that in Figure 4.11(c) and (d). However, a large number of non-salient pixels are detected as salient ones by the methods of [43, 68]. Thus samples that should be distributed to salient pixels are wasted on non-salient pixels that are detected as salient ones. In the close-up by using the proposed saliency detection method (Figure 4.11(g)), the quality of the tent is the highest (40.4dB), and we can see that the quality is consistent for the entire tent.

Figure 4.12 illustrates the perceptually rendered images for the Fairy scene by using different saliency detection methods. In the close-ups of the salient regions by using different saliency detection methods (Figure 4.12(c) - (g)), the close-up by using the method of [68] (Figure 4.12(e)) and the close-up by using the proposed method (Figure 4.12(g)) exhibit higher rendering quality than the close-ups by other methods, since the two methods can well detect the salient “fairy” and distribute more rendering computation to the character. In the close-ups by using the method of [34, 43, 47] (Figure 4.12(c), (d), (f)), noise is visible in the area covering the object of “fairy”, since some pixels in the character are not detected as salient ones by the relevant saliency detection methods and less computational resources are distributed to these pixels.

In summary, the proposed saliency detection method well detects the entire salient

objects, and it can be used to guide allocation of computational resources in perceptual rendering in a reasonable way: distributing more computational resources to the entire salient objects, and less resources to non-salient objects. In the rendered images, quality does not change within each single salient object, and the problem of quality change within a single salient object is addressed.

4.5 Concluding Remarks

To detect entire salient objects in computer synthesized images, we make use of the accurate object boundary information available from 3D models in computing contrast. In existing graphics saliency detection methods, saliency is estimated by measuring pixel-level contrast, and some low-contrast pixels in salient objects cannot be detected. Considering that objects can be accurately segmented with the boundary information, in the proposed method, features are extracted on a per-object basis and contrast is computed by measuring the feature dissimilarity among different objects. By computing object-level contrast, the boundaries of salient objects can be detected accurately and thus, the salient objects can be detected effectively. Experimental results from the proposed method and existing related saliency detection methods on eight test images show that the proposed method yields better performance in terms of precision, recall and F-measure indices.

In the proposed saliency detection method, we have only considered the color and intensity features. Currently, we have not incorporated depth information in the computation of saliency map. However, depth information is also an important saliency cue, and can be considered as a feature channel in the computation of the object-

level contrast. In the future work, we are to study how depth information influences visual saliency and to propose a saliency detection model which incorporates depth information.

Additionally, we adopt the proposed visual saliency detection method in the application of perceptual rendering. Experimental results show that incorporating the proposed saliency detection method in perceptual rendering helps to address the problem of inconsistent quality within a salient object in saliency-based perceptual rendering.

In Chapters 3 and 4, we have discussed issues related with the saliency-based perceptual rendering; however, there are some other perceptual properties of the HVS that have not been studied and exploited in the graphic rendering process. In the next chapter, we will study one such additional property: i.e. low sensitivity to irregular structure, and incorporate this property to benefit perceptual graphic rendering.

Chapter 5

Perceptual Rendering with Irregularity Measure

Rendering distortion is annoying, and a large amount of computational resources are required to alleviate the distortion. In this chapter, we explore how to decrease the rendering cost of 3D scenes which contain a large amount of irregular texture by making use of a property of the HVS, i.e. low sensitivity to irregular structure. Since the sensitivity of the HVS to distortion signals decreases in irregular image regions, we can accelerate the rendering process by decreasing computational resources in irregular texture.

Existing visual masking based perceptual rendering methods do not differentiate regular and irregular regions, and thus often over-estimate the masking potential of regular regions. In this chapter, we propose a new method to detect the irregularity level of image content. A filter is designed to calculate the weighted average of each pixel's neighboring pixels. After filtering, structural information is preserved while ir-

regular details are removed. Based on the difference between the original and filtered images, we can differentiate irregular regions from regular ones. We incorporate the irregularity measure in 3D rendering, by distributing less computational resources to more irregular image regions. Experimental results demonstrate that the proposed method is able to accelerate rendering, while avoiding error due to aggressively decreasing computational resources on regular regions.

5.1 Introduction

In Chapters 3 and 4, we explore the visual attention mechanism of the HVS and incorporate visual saliency into an adaptive sampling scheme for rendering acceleration. The perceptual rendering scheme proposed in Chapters 3 and 4 is suitable for general scenes since a scene usually contains one or more salient objects. In this chapter, we explore another property of the HVS, i.e. low sensitivity to irregular structure, for rendering of scenes which contain irregular textural information. Watson *et al.* [138] proposed the term “entropy masking” to reflect the notion that the masking is a function of the degree to which the mask is unknown. The less an observer knows a mask, the stronger the entropy masking effect of the mask will be. Watson *et al.* [138] also indicated that the degree to which a pattern is unknown is related with the complexity of the pattern, i.e. a complex pattern has strong masking effect than a simple pattern. An irregular pattern is with high complexity, thus an irregular pattern has strong masking effect. The property of the HVS can be used to accelerate rendering by decreasing computational cost in irregular pattern without introducing perceptually significant rendering errors.

Figure 5.1: Examples of masking properties of different image contents. The top row: high-quality rendered images; bottom row: low-quality rendered images. For the low-quality rendered images, we only decrease the computation on the teapot, but maintain the computation of the background regions. From (e)-(h), with higher irregularity in image content the rendering noise becomes less visible due to the masking of textural patterns.

Figure 5.1 illustrates the masking properties of different image contents. Four reference images are shown in Figure 5.1(a) - (d). Four low-quality rendered images are shown in Figure 5.1(e) - (h). In the four images (Figure 5.1(e) - (h)), the teapots are rendered with the same amount of computation, and thus have similar objective levels of rendering noise. As shown in this figure, the teapot texture in Figure 5.1(a) and (e) is smooth, the texture in Figure 5.1(b), (c), (f) and (g) is regular, while the texture in Figure 5.1(d) and (h) is irregular. Compared with the reference images, the noise in Figure 5.1(e) - (g) are noticeable, but it is difficult to detect the noise in Figure 5.1(h). From this example, we can see that noise becomes less visible as a result of the increased irregularity level of texture. The masking properties of image content can be used to accelerate graphic rendering by decreasing computational cost in irregular regions.

There are existing perceptual rendering methods concentrating exclusively on visual masking [104, 135]. These methods detect regions with strong masking ability and decrease rendering computation on these regions. The method [135] computes visual masking properties of image content by using components of JPEG compression standard. The assumption of that study is that at moderate compression rates, the quantization errors in JPEG compression are largely unnoticeable, i.e. the error between the original and JPEG decoded images is below a masking threshold of the image content. Visual masking potential is estimated as the difference between the original and JPEG decoded images.

Qu *et al.* [104] proposed to compute the visual masking properties of image content based on a computational model of human vision. The method does not use the vision model to predict the difference between two intermediate rendering results and update the vision model during rendering as other perceptual rendering methods [10, 107] do. Instead, the method uses the vision model to estimate the masking potential of image content. Two maps are generated: one indicating the visual difference between the original image and a filtered one; the other indicating the visual masking due to the threshold versus intensity function, by using piece-wise approximation developed by Ferwerda *et al.* [31]. The two maps are combined into a final map, which indicates the visual masking potential in the original image.

Existing perceptual rendering methods often assume strong masking ability for high contrast and rapid changing regions, and distribute less rendering resources to such regions. These methods do not differentiate regular regions from irregular ones. However, regular regions, including edges and regular textures, have lower levels of masking potential than irregular regions. Thus these methods may generate

perceivable errors on regular regions in rendered images. In the study in this chapter, we address the problem by distributing rendering resources based on the irregularity of image content.

We propose a method to estimate the irregularity level of image content. For regular regions, the differences between center pixels and their neighboring pixels are small; for irregular regions, the differences between center pixels and their neighboring pixels are large. Based on the fact, we design a filter to compute the weighted average of each pixel’s neighboring pixels in an image. The weight for each pair of pixels is determined by the similarity between the two pixels. After filtering, regular structure is preserved while irregular details are removed. The proposed method estimates the irregularity as the difference between the original image and the filtered one.

We incorporate the irregularity measure in an adaptive sampling scheme. The difference between this adaptive sampling scheme and the scheme proposed in Chapter 3 is that we replace the saliency detection module with the irregularity detection module. The proposed perceptually based adaptive sampling scheme generates a coarse image first and then computes an irregularity map that indicates the masking potential of the image content. In adaptive sampling, sample distribution is guided by the complexity levels as well as the irregularity levels of pixels. Compared with the non-perceptual adaptive sampling scheme, our adaptive sampling scheme is able to accelerate rendering by a factor about 2.6, for the test images in the experiments.

The proposed perceptual rendering method differs from existing perceptual rendering methods [10, 31, 89, 90, 91, 104, 107, 135, 144] which take into account the visual masking property. Existing perceptual rendering methods speed up rendering by decreasing the computation on regions with strong masking effect. These meth-

ods do not differentiate regular regions from irregular ones, and often produce high masking values for edges and some regular textures. As a result, a small amount of computational resources are allocated to such regular regions, including edges and regular texture in the rendering process. However, the aggressive decrease of computational resources in regular regions introduces visible noise, and this problem will be further discussed with examples in Section 5.3.1. With the proposed rendering method, we can avoid the visible noise in rendering.

The novelty of the proposed method is twofold. First, we design a method to detect irregular regions, which exhibit strong masking potential of rendering errors. Second, we incorporate the irregularity measure in a perceptually based adaptive sampling scheme, which speeds up rendering by adaptively distributing samples according to the estimated complexity levels and the irregularity levels of pixels.

5.2 Perceptual Rendering Based on Irregularity Measure

5.2.1 Existing Entropy Masking Algorithm

Some existing entropy masking methods [1, 83, 113] try to detect the irregularity of image content using Shannon entropy [115]. Given a pixel x_o , a luminance histogram is first constructed from pixels in the $m \times m$ local patch P_{x_o} centered at x_o . Then the irregularity level of pixel x_o is calculated as the entropy $H(P_{x_o})$ of patch P_{x_o} :

$$H(P_{x_o}) = - \sum_{i=0}^M \text{prob}(L_i) \cdot \log \text{prob}(L_i) \quad (5.2.1)$$

where $L_0, \dots, L_M$ are luminance values of pixels in the patch P_{x_o} , and $prob(L_i)$ is the occurrence rate of luminance value L_i in patch P_{x_o} . In Equation 5.2.1, the computed entropy of an image patch solely depends on the relative occurrence of pixel values in the patch regardless of the structural regularity of the patch. The spatial correlation between each center pixel and its neighbors is not considered, and as a result, patches of different levels of irregularity but the same histogram will result in the same entropy value. Therefore, Shannon entropy cannot detect the irregularity level of image content.

5.2.2 The Proposed Irregularity Measure

In smooth and regular patches, pixels are highly correlated with their neighboring pixels; while in irregular patches, the correlation between pixels and their neighboring pixels is low. The spatial correlation between pixels can be used to distinguish regular and irregular regions. To estimate the irregularity of images, we design a filter to compute the weighted average of each pixel's neighboring pixels. The weight assigned to each neighboring pixel is determined by the difference between the pixel and the center pixel. For regular regions, the structural information is preserved after filtering; while for irregular regions, the details will be removed in the filtered image.

The pipeline of the proposed method is shown in Figure 5.2. Starting from an input image, we use the luminance component (LAB color space) of the input image for irregular region detection, since the HVS is more sensitive to luminance component than to color components. For a pixel in the input image, a $(2R + 1) \times (2R + 1)$ neighborhood of the pixel is extracted, and difference d between the center pixel

Figure 5.2: The pipeline of the proposed irregularity detection scheme.

and each of its neighboring pixels is computed. Similarity (Sim) between each pair of center-neighboring pixels is calculated based on the difference. Then the filter computes the similarity-weighted sum of neighboring pixels, and replaces the center pixel with the sum in the filtered image. Finally, an irregularity map is computed based on the difference between the original image and the filtered one.

To effectively estimate the level of irregularity, regular regions are expected to remain unchanged and irregular details are expected to be removed after filtering. The simplest way to obtain the difference between each pair of center-neighboring pixels is to calculate the luminance difference $d_L(x_o, x_i)$:

$$d_L(x_o, x_i) = (L_{x_o} - L_{x_i})^2 \quad (5.2.2)$$

where L_{x_o} and L_{x_i} are the luminance values of the center pixel x_o and a neighboring pixel x_i , respectively. After the difference computation, each filtered pixel $L_{x_o'}$ is computed as a similarity-weighted sum of the neighboring pixels:

$$L_{x_o'} = \frac{\sum_{x_i \in \mathcal{N}(x_o)} \exp\left(-\frac{d_L(x_o, x_i)}{2\varsigma^2}\right) L_{x_i}}{\sum_{x_i \in \mathcal{N}(x_o)} \exp\left(-\frac{d_L(x_o, x_i)}{2\varsigma^2}\right)} \quad (5.2.3)$$

where $\mathcal{N}(x_o)$ is the $(2R+1) \times (2R+1)$ neighboring patch of x_o ; $\exp\left(-\frac{d_L(x_o, x_i)}{2\varsigma^2}\right)$ is the

Figure 5.3: Filtering results of two patterns using the luminance difference and the patch difference. (a) and (d) are extracted from Figure 5.1(c) and (d), respectively. Target pixels x_o are marked with small squares, and their local patches are marked with bigger squares. In each image, three pixels with similar luminance values to the target pixel are also marked, together with the corresponding local patches. (b) and (e) are the filtered results of the luminance components of (a) and (d), respectively, using the luminance difference; (c) and (f) are the filtered results of the luminance components of (a) and (d), respectively, using the patch difference.

similarity $Sim(x_o, x_i)$ between x_o and the neighboring pixel x_i ; ς is a user specified damping factor, and a higher value yields a stronger blurring effect. In the extreme case, when ς approaches positive infinity, every neighboring pixel will have an equal contribution to the filtered value of the center pixel, and the filter becomes a mean filter.

The luminance difference between two pixels does not reflect any structural information around the pixels. In fact, even in irregular regions, it is highly possible

that we can find neighboring pixels with similar luminance values to the center pixel, and the similarity-weighted averaging operation will not change much the value of the center pixel. Figure 5.3(b) and (e) show the filtering results of two patterns using the luminance difference. In Figure 5.3(a) and (d), x_o is highlighted with a small shadowed square and the $(2r + 1) \times (2r + 1)$ ($r < R$) local patch centered at x_o is highlighted with a larger shadowed square; several neighboring pixels which have similar luminance values to x_o are also highlighted, together with the corresponding $(2r + 1) \times (2r + 1)$ local patches. Note that in Figure 5.3(a), the local patterns of the other marked pixels are similar to that of the center pixel x_o , while in Figure 5.3(d), the local patterns of the other marked pixels are different from that of the center pixel x_o . According to the luminance difference, in Figure 5.3(a) and (d), the marked pixels are considered as similar to the center pixel, and are given large weights in filtering. Figure 5.3(b) and (e) show the filtered images of the luminance components of Figure 5.3(a) and (b), respectively, using the luminance difference. We can see that regular information is well preserved, but irregular visual information is not removed.

To differentiate regular regions from irregular ones, we also need to consider the difference between local patterns of the center and neighboring pixels. Here we use the simple MSE (mean square error) metric. The local patch distance $d_P(x_o, x_i)$ between two pixels is calculated as the average of the per-pixel squared distances over the $(2r + 1) \times (2r + 1)$ patches:

$$d_P(x_o, x_i) = \frac{1}{(2r + 1)^2} \sum_{x_n \in P_{x_i}} \sum_{x_m \in P_{x_o}} (L_{x_m} - L_{x_n})^2 \quad (5.2.4)$$

where L_{x_m} and L_{x_n} are the values of the pixels in patches P_{x_o} and P_{x_i} , respectively. To further improve the detection performance for patches with similar structure but

different mean values, we subtract each pixel value with the mean value of the local patch, and thus Equation 5.2.4 can be converted to:

$$d_P(x_o, x_i) = \frac{1}{(2r+1)^2} \sum_{x_n \in P_{x_i}} \sum_{x_m \in P_{x_o}} ((L_{x_m} - \overline{L_{P_{x_o}}}) - (L_{x_n} - \overline{L_{P_{x_i}}}))^2 \quad (5.2.5)$$

where $\overline{L_{P_{x_o}}}$ and $\overline{L_{P_{x_i}}}$ are the mean values of patches P_{x_o} and P_{x_i} , respectively. By using the patch difference, the neighboring pixels highlighted in Figure 5.3(d) will not be considered as similar to the center pixel x_o , and thus will not be given large weights during the filtering process. Figure 5.3(c) and (f) show the filtered images of the luminance components of Figure 5.3(a) and (d), respectively, using the patch difference. We can see that regular structural information is preserved and irregular details are removed after filtering.

Using the patch difference removes irregular information but will also blur certain regular regions, since even in regular regions, some pixels may not have exact matches in terms of patch difference, such as the highlighted pixel x_o on the curved lines in Figure 5.4(a). The local patch of the highlighted neighboring pixel x_i is not an exact match to the local patch of x_o , due to the slight differences in the directions of the curves. Note that the pixels on the curved lines are blurred after filtering as shown in Figure 5.4(b). Figure 5.4(c) shows the absolute difference between (a) and (b), and there is large difference between the original image and the filtered one.

To effectively estimate the level of irregularity, we use both the luminance difference and the patch difference in the filtering operation. A high weight for luminance difference is expected in strong edges to preserve the structural information, while a high weight for patch difference is expected in non-edge regions to remove the detailed visual information. Therefore, we adopt the variance $var(\cdot)$ of the local patch,

Figure 5.4: Filtering result of one regular pattern using the patch difference. (a) Luminance component of a regular texture extracted from Figure 5.1(b). (b) Filtered texture using patch difference. (c) Absolute difference between (a) and (b).

which is an effective indicator of strong edges, to determine the weights of luminance difference and patch difference:

$$d(x_o, x_i) = (1 - v) \cdot d_L(x_o, x_i) + v \cdot d_P(x_o, x_i) \quad (5.2.6)$$

$$v = \exp\left(-\frac{\text{var}(P_{x_o})}{\mu}\right) \quad (5.2.7)$$

where μ is a parameter which controls the decay rate of v , we find that most of the test images have a range of variance between 0 – 100, and $\mu = 80$ works well in most cases. That means for regions with variance of 55 (a medium variance value), the weights given to luminance difference and patch difference are roughly the same (about 0.5).

After the difference computation, each filtered pixel $L_{x_o'}$ is computed as a similarity-weighted sum of the neighboring pixels:

$$L_{x_o'} = \frac{\sum_{x_i \in \mathcal{N}(x_o)} \exp\left(-\frac{d(x_o, x_i)}{2\zeta^2}\right) L_{x_i}}{\sum_{x_i \in \mathcal{N}(x_o)} \exp\left(-\frac{d(x_o, x_i)}{2\zeta^2}\right)} \quad (5.2.8)$$

The same filtering process is applied on all pixels of the input image, and a filtered image is generated. A final irregularity map is computed based on the difference

Figure 5.5: Irregularity detection result of the Room scene. (a) The original image, four regions are highlighted with blue squares. (b) Computation results of irregularity, higher brightness indicates higher level of irregularity. (c) Close-ups of the highlighted regions in (a). (d) Corresponding irregularity levels of the regions in (c), higher brightness indicates higher irregularity.

between the original image and the filtered one. Here, we would like to emphasize that it does not make sense to say that a single pixel is “regular” or “irregular”, since regularity is a semi-local property [1], depending on the neighborhood of one pixel. However, in our perceptual rendering algorithm, we need an irregularity map to guide the sample allocation of every pixel in an image, thus we compute the “irregularity of each pixel” on the local patch centered at that pixel. For simplicity, we use $I(x_o)$ to denote the irregularity of the patch centered at x_o . For each pixel x_o in the original image, we compute the irregularity $I(x_o)$ as the difference between variance of the two $(2r+1) \times (2r+1)$ local patches P_{x_o} and $P_{x_o'}$ centered at pixels x_o in the original image and x_o' in the filtered image, respectively.

$$I(x_o) = (\text{var}(P_{x_o}) - \text{var}(P_{x_o'}))^2 \quad (5.2.9)$$

Figure 5.5 shows an example of the irregularity detection result of the Room

scene. Four different regions are extracted from the original image, and shown in Figure 5.5(c). These four regions correspond to four different types of regions in an image: smooth region, edge, regular texture and irregular texture. The corresponding irregularity values of these four regions are shown in Figure 5.5(d), and we can see that the smooth regions and regular regions (including edge and regular texture) exhibit low values, while the irregular texture exhibits higher values. With the proposed method, irregular regions are detected, and the detection results will be used in the proposed perceptual rendering algorithm.

5.2.3 Perceptually Based Adaptive Sampling

We incorporate the proposed irregularity measure in an adaptive sampling scheme. The main idea of the perceptually based adaptive sampling scheme is to decrease the total number of samples without degrading the visual quality of rendered images by distributing fewer samples to irregular image regions. Figure 5.6 shows the block diagram of the adaptive sampling scheme. The adaptive sampling scheme is similar to the one proposed in Chapter 3, and we replace the “saliency map” in Figure 3.3 with the “irregularity map” in Figure 5.6. In the adaptive sampling scheme, the image plane is sampled uniformly with a small number of initial samples at every pixel. A coarse image is rendered by computing the direct illumination of the sample rays (i.e. one sample ray experiences at most one bounce from the camera to light sources), as a quick estimation of the final image. Using direct lighting works well in most cases, however, one limitation is that regions lit mostly by indirect light will be almost black in the coarse image, and these regions will be detected as low-regularity

Figure 5.6: Block diagram of the proposed adaptive sampling scheme in path tracing.

ones. The penalty of possible misdetection is that the potential sample reduction will be decreased. This coarse image is subsequently used as input to the computation of an irregularity map.

The global illumination values of the initial sample rays are computed using the path tracing algorithm. Then the per-pixel rendering complexity is computed by the variance:

$$\varepsilon(x_o) = \sqrt{\frac{1}{V_{x_o}} \sum_{i \in Area(x_o)} (s_i - \bar{s}_{x_o})^2} \quad (5.2.10)$$

where $\varepsilon(x_o)$ is the estimated error of pixel x_o , V_{x_o} is the number of samples within the region $Area(x_o)$ in the pixel; s_i is the value of a sample for this pixel, and $\bar{s}_{x_o}$ is the average sample value of all the samples for pixel x_o . The estimated complexity level of each pixel is stored in a complexity map.

To exploit the masking of irregular regions in adaptive rendering, a priority map is generated as a combination of the irregularity map and the complexity map. Since irregular regions have higher noise masking potential, less numbers of samples are needed. The final priority map is calculated as:

$$\Theta(x_o) = \varepsilon(x_o) \cdot \exp\left(-\frac{I(x_o)}{2\gamma^2}\right) \quad (5.2.11)$$

Figure 5.7: The effect of different γ in sample distribution of the Room scene (Figure 5.5(a)). (a) Sample distribution map using only the complexity value ε . (b) Sample distribution map at $\gamma = 5$. (c) Sample distribution map at $\gamma = 3$. (d) Sample distribution map at $\gamma = 1$.

where $\Theta(x_o)$ is the priority value for pixel x_o , $\varepsilon(x_o)$ is the complexity value, $I(x_o)$ is the irregularity value; γ is a parameter which controls the decay rate of I . If γ is large, the complexity map will dominate the sample distribution. If γ is small, the irregularity map will have more influence. The sample distribution maps of the Room scene with different γ are illustrated in Figure 5.7. The sample distribution map by using only the complexity value ε is shown in Figure 5.7(a). The sample distribution maps using both the complexity value ε and the irregularity value I are shown in Figure 5.7(b) - (d). As γ decreases from 5 (Figure 5.7(b)) to 3 (Figure 5.7(c)), the influence of the irregularity map increases, i.e. more samples are distributed to irregular regions (the irregularity map is shown in Figure 5.5(b)). However, a too small γ value will remove almost all the samples on irregular regions, as shown in Figure 5.7(d). In the experiments, we have used $\gamma = 3$. The scheme allocates an equal proportion of the total sample budget in each iteration, and the per-pixel sample count is proportional to the pixel's priority value from previous iteration. After the iterative sampling process, a final image is generated from the sampled values using a filter, which

computes each output pixel as a weighted sum of nearby samples.

5.2.4 Discussion for Irregularity Measure Parameters

For each center pixel, the proposed irregularity filter computes a similarity-weighted average of its neighboring pixels. The patch size r in the patch difference computation and the neighborhood size R need to be determined. To determine the patch size, we have conducted experiments by using different patch sizes under a given neighborhood size. In the experiment, we have tried 3×3 ($r = 1$), 7×7 ($r = 3$) and 13×13 ($r = 6$) patch sizes under a neighborhood size $R = 6$ (we will show the effect of R in another experiment).

Figure 5.8 shows the difference maps between an original image and the filtered images at different patch sizes. We show the false-colored version of each map for better representation. In the maps, warmer color indicates a higher difference. The regions highlighted in Figure 5.8(b) and (c) correspond to an irregular region in the original image shown in Figure 5.8(a). It is expected that the irregular details are removed after filtering. It can be observed that the highlighted region in Figure 5.8(b) is darker than that in Figure 5.8(c) (i.e. the differences between the original image and the filtered image are smaller). That means, using 3×3 patch removes fewer irregular details than using 7×7 patch, and yields a too conservative estimation of the irregularity. The reason is that if the local patch size r is too small, spatial information around each pixel cannot be captured, and irregular details will not be removed after filtering. The region highlighted in Figure 5.8(d) corresponds to a regular texture on the poterie. It is expected that the regular structure be preserved

Figure 5.8: Comparison of different patch sizes. (a) Image of the Poterie scene. (b) Absolute difference between the original image and the filtered image using 3×3 local patch. (c) Absolute difference between the original image and the filtered image using 7×7 local patch. (d) Absolute difference between the original image and the filtered image using 13×13 local patch.

after filtering; however, using a 13×13 patch blurs the regular regions, and causes large deviations from the original image, as shown with orange and red spots in the highlighted regions in Figure 5.8(d). The reason is that if the local patch size is too large, it would be difficult to find matched patches for fine regular structure, and such regular regions will be removed after filtering. Based on the experiment, we used a local patch size 7×7 ($r = 3$) in the proposed irregularity measure.

We also need to determine the neighborhood size R . A large neighborhood allows the method to find matching pixels in regular regions, but increasing the neighborhood size also increases the complexity. Figure 5.9 shows the computational cost (in seconds) of the filtering step on the Teapot scene with different neighborhood sizes. It shows that as the neighborhood size R increases, the cost also increases significantly. Considering that our goal to include the irregularity measure is to accelerate rendering, the irregularity map computation has to be efficient; otherwise the use of such a map defeats its purpose. In the experiment, we have used a neighborhood size

Figure 5.9: Computational cost (in seconds) of the filtering step on the Teapot scene with different neighborhood sizes.

13×13 ($R = 6$), and this size allows the filter to capture the regular structure with acceptable overhead.

Another important parameter of the proposed filter is ς in Equation 5.2.8. As discussed in Section 5.2.2, ς is a user specified damping factor, and a higher value yields a stronger blurring effect. Figure 5.10 shows the filtered images using different ς . As ς increases from 5 (Figure 5.10(b)) to 20 (Figure 5.10(d)), more details are removed. A proper choice of ς also depends on image size, display resolution and viewing distance. For example, the bushes on the Plant scene in Figure 5.13 are considered as irregular details from large viewing distance, but they may be considered as regular structure if the viewing distance is sufficiently small to view every single leave in the bushes clearly. In the experiments, test images were rendered with resolution of 400×400 pixels, and shown on a 19-inch LCD monitor with display resolution at 1440×900 pixels and viewing distance at 60 cm. We have used $\varsigma = 5$ in the experiments and it works well for all the test images.

Figure 5.10: Filtered images of the Tribal scene using different ς . (a) Luminance component of the original image. (b) Filtered image at $\varsigma = 5$. (c) Filtered image at $\varsigma = 10$. (d) Filtered image at $\varsigma = 20$.

5.3 Experimental Results and Discussions

5.3.1 Comparison to Existing Visual Masking Methods in Perceptual Rendering

In this section, we compare the performance of the proposed irregularity measure with three existing visual masking computation methods [74, 104, 135] in perceptually based adaptive sampling. The method in [135] computes visual masking properties of image content using components of JPEG compression standard. The authors applied the visual masking computation method in an adaptive path tracer [135]. The study in [104] computes the visual masking potential using a human vision model, Visual Discrimination Metric (VDM) [79]. In this chapter, we replace the VDM with another widely used computational model VDP [20]. We use an implementation called HDR-VDP-2 (High dynamic range VDP) ¹. This is an open-source implementation, and has better performance than VDM [85]. Another method [74] has been

¹http://sourceforge.net/apps/mediawiki/hdrvdp/index.php?title=Main_Page

Figure 5.11: Comparison of the irregularity maps by the proposed method and the visual masking maps by existing visual masking computation methods. (a) Reference images. (b) Visual masking map by the method [135]. (c) Visual masking map by the method [104]. (d) Visual masking map by the method [74]. (e) Irregularity map by the proposed method. In (b) - (d), higher brightness indicates higher masking ability.

proposed for Just Noticeable Distortion (JND) detection in the area of image processing. A JND model computes the visibility threshold of visual signals, such as noise in an image. Given an image, the JND model [74] computes the visibility threshold of noise with respect to the luminance and contrast of image content. This visual masking computation method [74] has been successfully applied in perceptual quality evaluation, and perceptual image compression.

Figure 5.11 illustrates the irregularity maps by the proposed method and the visual masking maps by existing visual masking methods. Four categories of regions: smooth, edge, regular texture and irregular texture are marked in Figure 5.11(a). Maps in Figure 5.11(b) are generated by the method [135]. In these maps, smooth regions exhibit low values and irregular regions exhibit high values. However, edges

and regular texture also exhibit high values. The reason for such kind of results is that the method [135] is based on JPEG compression standard, and it always generates high values for high-frequency regions, including edges, and rapid changing regular/irregular regions. Maps in Figure 5.11(c) are generated by the method [104]. The method [104] makes use of a computational vision model, and it considers the contrast sensitivity function and contrast masking. In maps shown in Figure 5.11(c), regular texture exhibits slightly higher values than edges and irregular texture. Maps in Figure 5.11(d) are generated by the method [74]. Regular texture still exhibits higher values than irregular texture. Maps in Figure 5.11(d) are generated by the proposed method. We can see that the calculated values for irregular texture are high, and the values for smooth regions, edges, and regular texture are very low. The three existing methods all share the same problem: edge regions and rapid-changing regular regions are given high masking values. In fact, these regular regions do not have high masking degrees and distortions can be easily noticed. In comparison, our method well differentiates irregular regions from regular ones.

Figure 5.12 shows the images rendered with the perceptual rendering method by different visual masking schemes. The Teapot scene is shown at the top row. Four regions, including edge, regular texture and irregular texture, are amplified for a clearer comparison. We can see that visible noise appears on the edge and regular texture in Figure 5.12(b) - (d), since the test methods over-estimate the masking potential of these regions, and aggressively decrease the number of samples distributed to these regions. However, the proposed method does not have this problem and yields high perceived quality on both regular regions and irregular ones, as shown in Figure 5.12(e). The Poolball scene is shown at the bottom row. The numbers on

Figure 5.12: Comparison to existing visual masking schemes in perceptual image rendering. (a) Reference images. (b) Rendered images using the method [135]. (c) Rendered images using the method [104]. (d) Rendered images using the method [74]. (e) Rendered images using the proposed irregularity measure.

the balls are with high luminance contrast to nearby pixels, and also include a large amount of high frequency components; thus pixels covered by the numbers exhibit high masking values by the methods in [74, 104, 135]. However, decreasing samples on the ball number will introduce noise around the boundaries of the numbers, as shown in Figure 5.12(b) - (d). The proposed method estimates higher irregularity in the green table texture, and much lower irregularity in the regular regions. As a result, the proposed method yields good quality on such regular regions, as shown in

Figure 5.13: Test scenes used in the experiment.

Figure 5.12(e).

The examples above show that the proposed method distributes samples in a more effective way: fewer samples are given to irregular image regions and the saved samples are used to enhance the quality of regular regions. To further validate the rendering performance of proposed method and estimate the improvement quantitatively, we have conducted subjective viewing tests. The experiment setup and test results will be presented in Section 5.3.2 below.

5.3.2 Speedup Measurement

We compare the proposed method with a recently proposed non-perceptual adaptive sampling method [55]. Computational saving achieved by the proposed perceptual rendering method is estimated through an equal-quality comparison between the method [55] and the proposed method. Reference images are rendered at high sampling densities by the method [55], and test images are rendered at lower sampling densities by the proposed method. In this study, we have conducted a subjective

experiment to test whether the perceived quality of reference images and test images are the same.

Eight scenes (Fairy, Plant, Poolball, Poterie, Room, Sponza, Teapot and Tribal) are used in the subjective test, and the test scenes are illustrated in Figure 5.13. Note that all these test scenes contain a large amount of irregular textural information. Twenty-four participants were involved in the subjective test. All the participants are students or staff from the School of Computer Engineering, Nanyang Technological University, Singapore. The participants are with normal vision or corrected to normal vision. All the participants were naïve to the purpose of the experiment. The subjective viewing test was performed in normal office lighting. Test images were shown in a 19 inch LCD display and the viewing distance was set to 0.5m. Before the formal test, a short introduction was given on the test procedure. In the formal experiment, one group of images were shown on the screen each time: one reference image, rendered by the adaptive sampling method in [55], on the top of the screen and two images on the bottom of the screen. One of the two images is the same as the reference image, and the other one is the test image rendered by the proposed method at lower sampling densities. Participants performed 2AFC (two alternatives forced choice) tasks between the two bottom images, i.e. picking up one image which they thought was more similar to the reference image.

Table 5.1 shows the experimental results of the subjective viewing test. As can be seen, about a half (47% on average) of the participants thought that images rendered by the proposed method were more similar to the reference images than the reference images themselves, while approximately the other half of the participants thought that the reference images are more similar to the reference images themselves. For each

Table 5.1: The numbers of viewers in favor of images rendered by the two algorithms
(24 participants in total)

Scene	Ref	Proposed
Fairy	11	13 (54%)
Plant	14	10 (41%)
Poolball	12	12 (50%)
Poterie	11	13 (54%)
Room	11	13 (54%)
Sponza	20	4 (17%)
Teapot	9	15 (62%)
Tribal	13	11 (46%)

pair, a result of 50% to 50% (12/12) selection is the expected result when no noticeable differences exist between the reference images and the images by the proposed method.

We have further performed statistical analysis on the results of the subjective test. Table 5.2 shows the results of the χ^2 analysis. In this table, χ^2 value resembles a normalized sum of squared deviations between observed and expected frequencies (12/12), and p is the probability of getting a number equal or bigger than the observed one, assuming the null hypothesis (reference images and test images by the proposed method are chosen with equal probability). It can be seen that there is no significant difference between the reference images and the test images from the proposed method ($p > 0.05$) for most scenes, except the Sponza scene ($p = 0.001$). It indicates that the results shown in Table 5.1 are what we expected; that is, the user preference of images in terms of quality is by chance. Based on the experimental results in Table 5.1 and

Table 5.2: Output for a χ^2 test	
Scene	
Fairy	$\chi^2 = 0.167, p = 0.683$
Plant	$\chi^2 = 0.667, p = 0.414$
Poolball	$\chi^2 = 0.000, p = 1.000$
Poterie	$\chi^2 = 0.167, p = 0.683$
Room	$\chi^2 = 0.167, p = 0.683$
Sponza	$\chi^2 = 10.667, p = 0.001$
Teapot	$\chi^2 = 1.500, p = 0.221$
Tribal	$\chi^2 = 0.167, p = 0.683$

Table 5.2, the participants were not able to distinguish the test images rendered by the proposed method at lower sampling rates from the reference images rendered at higher sampling rates, for seven test scenes (except the Sponza scene).

Figure 5.14 illustrates the visual difference maps between the test images and the reference images for the Sponza scene, the Teapot scene and the Fairy scene, by using HDR-VDP-2 [85]. HDR-VDP-2 predicts the probability of the differences between two images visible to the human observer. It takes two images as input and generates a map in which each pixel value represents the probability of detecting a pixel difference between the two images by a typical human observer. Note that HDR-VDP-2 does not differentiate irregular regions from regular ones in the masking computation, but it can provide useful information on whether or not the visual errors appear on irregular regions, where the noise masking potential is high.

For the Sponza scene, the visualized difference map in Figure 5.14(c) shows that

Figure 5.14: HDR-VDP-2 visual differences between the reference images and the test images by the proposed method. (a) Reference images. (b) Rendered images using the proposed method. (c) Visual difference map (warmer color indicates higher possibility that a human observer detects a pixel difference between the reference image and the test image).

visible errors present on both regular regions and irregular regions. The reason is that some low-regularity regions in this scene (e.g. shadow areas highlighted with rectangles) are lit mainly by indirect illumination and pixels of these regions are with relatively slow convergence rates. The saved samples from irregular regions cannot increase the perceived quality of these regular regions effectively. Since quality decrease is easy to be noticed on regular regions, most participants (83%) thought that the perceived quality of the image rendered by the proposed method is worse than that of the reference image for the Sponza scene. For the Teapot scene, most visual differences (regions with warmer colors in Figure 5.14(c)) reside on irregular regions. As shown in the results in Table 5.1, participants in our experiments do not notice the difference for the Teapot scene (62% of the participants can not pick up the reference image from the two bottom images). Similarly, for the Fairy scene, most visual differences reside on irregular regions. As shown in the results in Table 5.1, more than half of the participants (54%) in our experiments do not notice the difference for the Fairy scene.

We record the computational cost (in seconds) for the test images and the reference images in Table 5.3. The test images and the reference images are rendered on a server computer with Intel X5470 CPU at 3.33GHz. The overall cost of the proposed scheme is the sum of three components: coarse image generation, irregularity map computation, and the main rendering process. Since the computational cost on coarse image synthesis and irregularity map computation is much less than the saved cost from sample reduction, the overall rendering process has been accelerated through integration of the irregularity measure into adaptive sampling. Speedup is calculated by comparing the rendering cost of reference images and the overall cost by the

Table 5.3: Cost (in seconds) for the reference images and test images by the proposed method.

	Ref	Proposed				
Scene	overall time	overall time	render time	rough image	irreg. map	speedup
Fairy	3058.8	519.7	476.2	39.0	4.5	5.9
Plant	4338.3	1192.0	1042.7	143.8	5.5	3.6
Poolball	3150.4	1448.6	1439.3	3.8	5.5	2.2
Poterie	401.2	199.3	193.4	1.3	4.6	2.0
Room	1240.7	699.5	690.3	3.5	5.7	1.8
Teapot	564.6	512.1	480.7	25.8	5.6	1.1
Tribal	518.3	333.1	300.1	27.6	5.4	1.6

proposed method.

The computational speedup ranges from 1.1 to 5.9 (2.6 on average). In general, the computational saving depends on image content, e.g. higher saving can be obtained for a highly textured scene than a smooth scene. In addition, the time spent on tracing a sample ray varies at different locations on an image. In the extreme case, if most irregular regions of a scene are lit by direct lighting (where the average cost of tracing a ray is low) while most regular regions of the scene are lit by indirect lighting (where the average cost of tracing a ray is high), even removing all samples from the irregular regions will not yield much speedup, since the original computational cost of the irregular regions in the scene is negligible. The worst case occurs on the Teapot scene, and the cost of the reference and test images is almost the same. The reason

is that tracing rays hitting on the teapot takes longer time than tracing rays hitting on the table in this scene, and the proposed method reduces the samples on the table and moves parts of the saved samples to the teapot. Thus, although the overall sample number decreases, the overall cost does not decrease significantly. The best case occurs on the Fairy scene, with the achieved speedup at 5.9. The reason is that this scene contains a lot of irregular regions which are lit by indirect lighting. As the sample number decreases on indirect-lighted regions, the computational overhead decreases even faster. Overall, experimental results show that incorporating irregularity information in adaptive image rendering leads to significant computational saving without degrading the perceived quality of resultant images.

5.4 Concluding Remarks

In this chapter, we have designed and implemented a method to estimate irregularity of images and incorporated the irregularity measure to guide sample allocation in a perceptually based adaptive sampling scheme for rendering of scenes containing a large amount of irregular texture. In our scheme, irregularity of image content is estimated based on the luminance and patch similarity between pixels and their neighbors. The proposed method can differentiate irregular texture from structurally regular regions, i.e. edges and regular texture. We have incorporated the irregularity detection scheme into an iterative adaptive sampling algorithm: the allocation of samples is guided by both the estimated complexity level and the irregularity of image content. Compared with existing visual masking based rendering methods, we can avoid the problem of over-estimating the masking potential of edges and regular

texture.

We conduct experiments to test the performance of the proposed scheme in rendering of scenes which contain irregular textural information. Experimental results have demonstrated that the proposed perceptual image rendering can get better rendering results compared with other visual masking based rendering methods. Additionally, the proposed perceptual image rendering method can obtain much time-saving without loss in perceived quality compared with the existing adaptive sampling method.

In Chapters 3 - 5, we introduce our work on perceptual graphic rendering, and in the next chapter, the work related with perceptual graphic quality evaluation will be described.

Chapter 6

Perceptual Quality Assessment for 3D Meshes

3D models are widely represented by triangle meshes, due to their simplicity. 3D meshes are subject to various visual distortions during geometry processing and transmission. In this study, we propose a novel objective quality assessment method for 3D meshes based on curvature information. According to characteristics of the HVS, two new components including visual masking modulation and saturation effect modulation are designed for the proposed method. Besides, inspired by the fact that the HVS is sensitive to structural information, we compute the structural distortion for 3D meshes. We test the performance of the proposed method on three publicly available databases of 3D mesh quality evaluation, and compare the performance of the proposed method with the relevant state-of-the-art methods. Experimental results demonstrate that the proposed method can predict consistent results in alignment with human perception across these databases.

6.1 Introduction

As aforementioned, in this chapter, we investigate issues on perceptual quality evaluation for 3D models represented by triangle meshes. Almost all mesh processing procedures cause some more or less visible changes in the appearance of 3D meshes, e.g. some mesh watermarking algorithms embed watermark information by modifying the length of the vectors that link vertices to the center of the model. In many cases, such as mesh watermarking, simplification or lossy compression, the goal is to make the change in appearance negligible, or as small as possible. To achieve this goal, a measure of the amount of distortion is needed. Traditionally, distortion is evaluated by geometric distance such as Hausdorff Distance (HD) and Root Mean Square Error (RMS); however, these metrics do not correlate well with the human judgement. In many graphics applications, the ultimate evaluator is a human observer, thus the distortion measure should be able to measure the perceived change in appearance by a human observer.

In this part of our study, we focus on quality evaluation of 3D triangle meshes based on curvature information. Curvature describes visual characteristics of 3D surfaces and has been studied in existing quality evaluation methods [64, 66, 128, 129, 136]. Given a reference model and a distorted model, the quality of the distorted model is measured through curvature comparison between vertices of the distorted model and the reference model. In the proposed method, the per-vertex difference in curvature is computed between the reference model and the distorted model, and the differences are pooled together into a distortion score for the distorted model. The distorted meshes studied in this method are with the same connectivity with the ref-

erence meshes, and extending the method to distortions which alter the connectivity of meshes, such as simplification, will be our future work.

Since the HVS is the ultimate evaluator of most produced 3D triangle meshes after rendering, an objective quality assessment method which correlates well with human perception needs to be designed. Existing studies have demonstrated that incorporating perceptual properties of the HVS will benefit various 3D multimedia applications, such as 3D model transmission [16], 3D model watermarking [63], 3D mesh simplification [104], and perceptual 3D rendering [23, 43]. According to characteristics of the HVS, we design two new modules, the visual masking modulation component and the saturation modulation component, in the proposed method to modulate the per-vertex curvature difference. Besides the visual masking and saturation modulation components, we also propose a method to compute the structural distortion of a distorted model and include this component in the proposed metric to account for the fact that the HVS is sensitive to structural information [137].

Visual masking refers to the decrease of visibility of visual signals with the presence of background signals [20]. In the context of 3D mesh quality evaluation, visual masking indicates that distortions are less visible on rough regions. Visual masking has been applied in 3D mesh quality evaluation in existing work [64, 66, 128, 129, 131, 136]. However, there are some problems in these modulations.

In the visual masking components in [64, 66], distortions occurring in low-curvature regions are over-estimated. In [64, 66], the difference D' in curvature between a vertex of the distorted model v_j^d and a vertex of the reference model v_i^r is expressed as a

contrast:

$$D'(v_i^r, v_j^d) = \frac{|\kappa(v_i^r) - \kappa(v_j^d)|}{\max(\kappa(v_i^r), \kappa(v_j^d)) + K} \quad (6.1.1)$$

where $|\cdot|$ is the absolute value; κ is the curvature value; K is a small constant added to avoid division by zero. The visual masking effect is considered in the above formulation, where D' is indeed the masked curvature difference of the original curvature difference $|\kappa(v_i^r) - \kappa(v_j^d)|$, and D' is in the range of $[0 - 1]$. In high-curvature regions, the curvature difference is masked to a large extent. There exists a problem with the formulation in Equation 6.1.1. When $|\kappa(v_i^r) - \kappa(v_j^d)|$ is close to 0, the curvature difference should be close to 0. However, if $\max(\kappa(v_i^r), \kappa(v_j^d))$ is also small, Equation 6.1.1 would give a value much larger than 0, and vertex distortion on low-curvature vertices is overestimated. To address this problem, we modulate the curvature change in a reasonable way, the details of which will be described in Section 6.2.3.

In visual masking components of the metrics in [128, 129, 131, 136], distortions are modulated by certain roughness (smoothness) measures. The roughness measure used in [136] is the Laplacian of Gaussian curvature, which is defined in 1-ring (nearest) neighbors of a center vertex. The smoothness measure defined in [131] is the dihedral angle between two adjacent triangles. The roughness measure defined in [129] is the Laplacian of the mean curvatures in 1-ring neighborhood of a vertex. The roughness measure in [128] is also defined in 1-ring neighborhood of each vertex. In the above mentioned visual masking components, the roughness (smoothness) measures are computed from the nearest neighbors of a vertex or triangle. There are a large number of vertices in a typical model. The variation in a small neighborhood is usually not significant, and thus it may be inaccurate to determine whether a region is

rough or not solely based on the statistics extracted from a small neighborhood. To overcome these problems, we propose a new visual masking modulation component for 3D meshes in this study. The proposed visual masking modulation component modulates the curvature difference with a roughness measure computed in a larger neighborhood.

Saturation effect in visual quality evaluation was first introduced by Lambrecht *et al.* [130]; this effect indicates that the sensitivity of the HVS to distortions decreases at high distortion levels. That is, when the distortion level is high, it would be difficult for human observers to perceive further increase in distortion level. Saturation effect has been considered in various quality evaluation methods for natural images and videos, such as [73, 93, 146]. However, saturation effect has not been considered in quality evaluation of 3D meshes. In this study, we investigate the saturation effect in visual quality assessment for 3D meshes and model the saturation effect with an exponential function.

The main contributions of this study include three aspects. First, we design a new component for visual masking modulation in quality assessment of 3D meshes. Second, we design a new component in the proposed method to account for the saturation effect in the HVS. Third, we include a component that computes the structural distortion of 3D meshes, with which the smoothing distortion can be well evaluated.

We conduct experiments to demonstrate the performance of the proposed method on three publicly available databases of 3D mesh quality evaluation: the LIRIS/EPFL general-purpose database [66], the LIRIS masking database [63], and the UWB compression database [131]. Experimental results demonstrate that the proposed method

Figure 6.1: The pipeline of the proposed 3D mesh quality assessment method.

yields consistent results with human perception across these databases.

6.2 The Proposed Quality Evaluation Method

6.2.1 Overview

The pipeline of the proposed method is shown in Figure 6.1. The proposed method relies on both the mean curvature difference and the structural distortion. Given a reference model and a distorted model, the proposed method calculates the mean curvature value (κ_{mean}) for each vertex in the models, and then computes an average value of mean curvature values ($\tilde{\kappa}_{mean}$) in the neighborhood of each vertex. After that, the proposed method computes the difference in average curvature values ($\tilde{\kappa}_{mean}$) between each pair of vertices in the two models. The computed curvature difference (variation) is then modulated by the visual masking component and the saturation component. Additionally, the method computes the structure dissimilarity between the two models to measure the distortion on sharp edges. Finally, a distortion score is derived through integrating the modulated curvature variation and the structural distortion.

6.2.2 Computation of Mean Curvature Difference

Intuitively, curvature describes the amount of deviation of a surface from being flat. Discrete curvature has been considered in several existing methods of 3D mesh quality assessment [64, 66, 128, 129, 136], since it well describes the visual characteristics of 3D meshes [66]. Figure 6.2 illustrates two mean curvature maps for a reference Bunny model, and a distorted Bunny model. The model in Figure 6.2(c) is created by applying coordinate quantization on the original model in Figure 6.2(a). Coordinate quantization is used in 3D mesh compression. In the rectangular coordinate system, each vertex in a 3D mesh is represented by three coordinates, x , y and z . Quantization truncates the vertex coordinates to a desired accuracy and maps them into integers that can be represented with a limited number of bits. Quantization can reduce the storage cost of vertex coordinate. The quantized 3D mesh shown in Fig. 6.2(c) is from the UWB compression database [131]. In Figure 6.2(b) and (d), warmer colors indicate higher mean curvature values. After the quantization, the surface of the Bunny becomes more rough from Figure 6.2(a) to (c), and the mean curvature map becomes more noisy from (b) to (d). We can see that mean curvature information is variant to surface distortion.

3D mesh is a piecewise-linear approximation of continuous surface, and there are various definitions of curvature tensor on a mesh [22]. Similar as in [64], we adopt the method proposed by Alliez *et al.* [2], to estimate the curvature tensor at each vertex of the mesh. The method estimates the curvature tensor on a geodesic disk neighborhood around each vertex, and yields stable results independent of the sampling density of the model [64]. More specifically, given a vertex v , and a neighborhood $\mathcal{N}$ which

Figure 6.2: Mean curvature describes the visual characteristics of 3D models. (a) Reference Bunny model. (b) Mean curvature map of the object in (a). (c) A distorted model after quantization of coordinates. (d) Mean curvature map of the object in (c).

approximates a geodesic disk around this vertex, the tensor $\mathcal{T}$ is calculated as:

$$\mathcal{T} = \frac{1}{Area(\mathcal{N})} \sum_{e \in \mathcal{N}} \varphi(e) \cdot |e \cap \mathcal{N}| \cdot \bar{e} \times \bar{e}^t \quad (6.2.1)$$

where e is each edge in the neighborhood $\mathcal{N}$, illustrated in Figure 6.3; $Area(\mathcal{N})$ is the surface area of the neighborhood $\mathcal{N}$; $\varphi(e)$ is the signed angle between the normals to the two triangles incident to e ; $|e \cap \mathcal{N}|$ is the length of the part of edge e lying within the neighborhood $\mathcal{N}$; for edges which intersect with the boundary of the neighborhood, $|e \cap \mathcal{N}| < |e|$; $\bar{e}$ is the unit vector of e ; $\bar{e}^t$ is the transpose of $\bar{e}$. With $\mathcal{T}$, the normal at vertex v is computed as the eigenvector of $\mathcal{T}$ with the minimum eigenvalue. The other two eigenvalues are defined as the maximum curvature κ_{max} and the minimum curvature κ_{min} . The mean curvature κ_{mean} is computed as $(\kappa_{max} + \kappa_{min})/2$.

For each vertex, the average value of mean curvature ($\tilde{\kappa}_{mean}$) is computed in a spherical neighborhood around the vertex. We adopt the Gaussian-weighted averages of curvature proposed in [64], and the average of mean curvature $\tilde{\kappa}_{mean}(v)$ of a vertex

Figure 6.3: Quantities used in computation of curvature tensor.

v is computed as:

$$\tilde{\kappa}_{mean}(v) = \frac{\sum_{v_m \in \mathcal{S}(v)} \exp(-2\|v_m - v\|^2/\phi^2) \cdot \kappa_{mean}(v_m)}{\sum_{v_m \in \mathcal{S}(v)} \exp(-2\|v_m - v\|^2/\phi^2)} \quad (6.2.2)$$

where v_m is a vertex in the spherical neighborhood $\mathcal{S}$ of v ; $\kappa_{mean}(v_m)$ is the mean curvature value of the vertex; ϕ is the radius of the neighborhood; $\|v_m - v\|$ is the Euclidean distance between vertex v_m and v . Here we compute the average curvature in a spherical neighborhood instead of a 1-ring neighborhood for the reason that curvature computed in a larger neighborhood is a better indicator of surface roughness, which will be demonstrated with an example in Figure 6.5. The difference in the mean curvature values between a vertex in a distorted model and the corresponding vertex in a reference model is calculated as follows:

$$D_o(v_i^r, v_j^d) = |\tilde{\kappa}_{mean}(v_i^r) - \tilde{\kappa}_{mean}(v_j^d)| \quad (6.2.3)$$

where v_j^d is a vertex in the distorted model M_d ; v_i^r is a corresponding vertex in the reference model M_r ; $\tilde{\kappa}_{mean}(v_i^r)$ and $\tilde{\kappa}_{mean}(v_j^d)$ are the average of mean curvature values of v_i^r and v_j^d , respectively; $|\cdot|$ computes the absolute value.

Figure 6.4: An example of visual masking effect. (a) Original LionVase model. (b) Distorted mesh created by adding medium-level noise in smooth regions. (c) Distorted mesh created by adding high-level noise in rough regions.

6.2.3 Visual Masking Modulation

Visual masking is an important characteristic of the HVS. In the context of quality assessment of 3D meshes, visual masking indicates that artifacts become less visible on rough regions. Figure 6.4 illustrates an example of visual masking effect. Figure 6.4(a) is the original LionVase model; two examples of smooth region and rough region in this model are highlighted. Figure 6.4(b) is a distorted model created by adding medium-level noise in smooth regions, and Figure 6.4(c) is a distorted model created by adding high-level noise in rough regions. The three models are extracted from the LIRIS Masking Database [63] (the details of this database will be described in Section 6.3.1). Although the objective level of distortion in Figure 6.4(c) is higher than that of Figure 6.4(b), the perceived quality in Figure 6.4(c) is better, since noise in the mane of the Lion is less annoying than noise in smooth regions of the face.

To take into account the visual masking effect for quality assessment, we mask the

curvature variation in rough regions. Existing methods [128, 129, 131, 136] calculate the roughness measure based on statistics extracted from a 1-ring neighborhood of a vertex or triangle. However, there are a large number of vertices in a typical 3D model and thus features of vertices, such as geometry coordinate, curvature, or dihedral angle, do not vary significantly in the 1-ring neighborhood. Due to the small variation, it is difficult to determine whether a local region is rough or not by observing the variation.

Figure 6.5(a) shows the mean curvature map for the LionVase model, calculated in 1-ring neighborhood of each vertex, and warmer colors indicate higher mean curvature values. As can be observed from Figure 6.5(a), although the vertices in the highlighted rough region exhibit higher curvature values than the vertices in the highlighted smooth region, in general, vertices are with low curvature values (dark blue colors). Some vertices in the rough region (e.g. the pixels highlighted with the circle in the rough region) have similar curvature values as vertices in the smooth region, and the masking potential of these vertices will be under-estimated.

In this work, we use the average curvature ($\tilde{\kappa}_{mean}$) of vertices computed in Equation 6.2.2 as the roughness measure. The average of mean curvature is calculated in a large spherical neighborhood of each vertex. A proper choice of radius of the neighborhood depends on display resolution, viewing distance and the sampling density at different parts of each model. For example, local regions of the mane in the LionVase model in Figure 6.4(a) are considered as rough from large viewing distance, but they may be considered as smooth if the viewing distance is sufficiently small to view every vertex clearly. In this work, the models are shown on a 19-inch LCD monitor with display resolution at 1440×900 pixels, and we set the radius to 0.2% of the max

Figure 6.5: Mean curvature maps for the LionVase model, (a) calculated in 1-ring neighborhood of each vertex, (b) calculated in a larger neighborhood as Equation 6.2.2.

length of the bounding box of the model empirically. An example of the curvature map is illustrated in Figure 6.5(b). From this figure, we can see that vertices in the rough regions are with high average curvature values (orange colors), while vertices in the smooth regions are with lower average curvature values (dark blue colors). Compared with Figure 6.5(a), rough region can be better distinguished from smooth regions in Figure 6.5(b).

To reduce the curvature variation in rough regions, we multiply the curvature variation $D_o(v_i^r, v_j^d)$ in Equation 6.2.3 by a masking function. The visual masking tuned curvature difference $D_{masking}(v_i^r, v_j^d)$ is calculated as follows:

$$D_{masking}(v_i^r, v_j^d) = D_o(v_i^r, v_j^d) \cdot \exp\left(-\frac{\tilde{\kappa}_{mean}(v_i^r)^2}{2\sigma^2}\right), \quad (6.2.4)$$

where σ is a parameter which controls the slope of the masking function. With the modulation, the curvature difference of low-curvature vertices will not be modified significantly (the masking coefficient is close the value “1”), to account for the weak

masking ability of these vertices. The computed curvature difference of high-curvature vertices will be reduced to a larger extent (the masking coefficient is significantly smaller than the value “1”), to account for the strong masking ability of these vertices.

After the modulation, distortions in rough regions will be masked. Take the two models in Figure 6.4(b) and (c) as examples. In the database [63], observers give a score in the range between 0 (worst quality) and 4 (best quality) to each distorted model. The mean opinion score (MOS) assigned to Figure 6.4(b) and Figure 6.4(c) is 0.644 and 2.041, respectively. That means the observers thought that Figure 6.4(b) has worse visual quality than Figure 6.4(c). The MOSs, the curvature differences, and the modulated curvature differences are listed in Table 6.1. The difference in curvatures of Figure 6.4(b), 22.8, is smaller than that of Figure 6.4(c), 31.5. That means the computed distortion level of Figure 6.4(b) is lower than that of Figure 6.4(c), and that is contradictory to the subjective assessment result. After the visual masking modulation, the computed distortion level of Figure 6.4(b), 11.9, is larger than that of Figure 6.4(c), 3.8, and this result is consistent with the subjective assessment result.

Table 6.1: MOSs, curvature difference, and modulated curvature difference of two distorted models

Type	Figure 6.4(b)	Figure 6.4(c)
MOS	0.644	2.041
Curvature Difference	22.8	31.5
Modulated Curvature Difference	11.9	3.8

Figure 6.6(a) presents the scatter plots of the MOSs versus the curvature difference for the LionVase model from the LIRIS masking database [63]. In this plot, each point represents one distorted LionVase model. Figure 6.6(b) presents the scatter

Figure 6.6: Scatter plots of the MOSs versus the curvature difference for the LionVase model (a) without visual masking modulation, (b) with visual masking modulation.

plots of the MOSs versus the visual masking modulated curvature difference for the LionVase model. The five-parameter logistic mapping curves between the curvature differences and the MOSs [the logistic mapping is to be discussed in Section 6.3.2 as Equation 6.3.1] are also shown in Figure 6.6(a) and Figure 6.6(b). With the visual masking modulation, the curvature differences correlate better with the MOSs.

6.2.4 Saturation Effect Modulation

Saturation effect indicates that sensitivity to perceived distortion decreases at high distortion levels [130]. Figure 6.7 presents an example of the saturation effect. This figure includes one reference model and two distorted models created by smoothing operation on the original model. The Taubin smoothing algorithm [125] is used to generate the distorted models in Figure 6.7(b) and Figure 6.7(c). The parameter “smoothing steps” controls the level of smoothing of the model. A higher step number induces larger blur in the models. The step number used in Figure 6.7(b) is 20 and the step number used in Figure 6.7(c) is 30. Thus, the model in Figure 6.7(c) has

Figure 6.7: Example on saturation effect. (a) Original RockerArm model. (b) A distorted model created by applying Taubin smoothing with step number of 20. (c) A distorted model created by applying Taubin smoothing with step number of 30.

higher level of distortion than the model in Figure 6.7(b). However, the perceived distortion levels in the two models are nearly the same.

We model the saturation effect with an exponential as Figure 6.8. The mapped curvature difference D_{sat} is calculated as follows:

$$D_{sat}(v_i^r, v_j^d) = 1 - \exp(-D_{masking}(v_i^r, v_j^d)/\beta) \quad (6.2.5)$$

where β is the Gaussian kernel. To illustrate how the plot in Figure 6.8 simulates the saturation effect, we show four distorted models, M_1 , M_2 , M_3 and M_4 , of a same reference model, as an example. In Figure 6.8, $D(M1)$, $D(M2)$, $D(M3)$ and $D(M4)$ denotes the curvature difference (distortion level) for distorted models M_1 , M_2 , M_3 and M_4 , respectively, before saturation modulation. M_3 is with a larger distortion level than M_1 (i.e. $D(M3) > D(M1)$). $\Delta D(M_1, M_2)$ is the difference in distortion level between M_1 and M_2 ; $\Delta D(M_3, M_4)$ is the difference in distortion level between M_3 and M_4 ; $\Delta D(M_1, M_2) = \Delta D(M_3, M_4)$. $\Delta D_{sat}(M_1, M_2)$ is the difference in distortion level between M_1 and M_2 after saturation modulation; $\Delta D_{sat}(M_3, M_4)$ is the difference in distortion level between M_3 and M_4 after saturation modulation.

Figure 6.8: Saturation effect modulation function.

We can see that $\Delta D_{sat}(M_3, M_4) < \Delta D_{sat}(M_1, M_2)$, which means that the sensitivity to perceived distortion decreases at high distortion levels.

After the saturation modulation, the pooled curvature difference between a reference model M^r and a distortion model M^d is calculated as:

$$D_{diff}(M^r, M^d) = \frac{1}{|M^r|} \cdot \sum D_{sat}(v_i^r, v_j^d) \quad (6.2.6)$$

where $|M^r| = |M^d|$ is the total number of vertices in each model.

6.2.5 Structural Distortion

In LIRIS/EPFL general-purpose database [66], there are two types of distortion: random noise and smoothing. Smoothing operation causes blur on the structure of 3D models. Since the HVS is sensitive to structural distortion [137], the blur of structure is annoying to human observers. We find that the difference in mean curvature is less effective in measuring the smoothing distortion, and the structural distortion is underestimated. Figure 6.9 shows a noise distorted model and two smoothing

Figure 6.9: Noise distorted and smoothing distorted models of the Venus. (a) Original Venus model. (b) A distorted model created by adding random noise. (c) A distorted model created by applying Taubin smoothing with step number at 20. (d) A distorted model created by applying Taubin smoothing with step number at 30.

distorted models. Figure 6.10(a) illustrates the scatter plot of MOSs versus predicted scores by the proposed method based on mean curvature difference, for the models in Figure 6.9(b), (c) and (d). In this plot, MOS is in the range between 0 (worst quality) and 10 (best quality). We can see that the rank of predicted score is not consistent with the rank of MOS and the smoothing distortion is underestimated.

To capture structural distortion of 3D meshes, we need to detect edges of the models and measure the distortions of edges. Since sharp edges are with high maximum curvature (κ_{max}) values, we detect vertices with high maximum curvature values and include these vertices in computation of structural distortion. An example of the maximum curvature map of the Venus model in Figure 6.9(a) is illustrated in Figure 6.11. In this figure, sharp edges and vertices are with warmer colors (higher κ_{max} values) and smooth vertices are with cooler colors (lower κ_{max} values). To compute the structural distortion, we compute the difference in κ_{max} values between vertices with high κ_{max} values in the reference model and the corresponding vertices in the

Figure 6.10: Scatter plots of MOSs versus predicted score by the proposed method for the meshes in Figure 6.9(b), (c) and (d). (a) MOSs versus predicted scores based on mean curvature difference. (b) MOSs versus predicted scores based on both mean curvature difference and structural distortion.

Figure 6.11: Maximum curvature map of the Venus model.

distorted model, more specifically, the structural distortion is computed as follows:

$$D_{struct}(M^r, M^d) = \frac{1}{N} \sum_{\kappa_{max}(v_i^r) > T_r} |\kappa_{max}(v_i^r) - \kappa_{max}(v_j^d)| \quad (6.2.7)$$

where T_r is a parameter determining which vertices are included in the computation of structural distortion; N is the number of vertices, which satisfy $\kappa_{max}(v_i^r) > T_r$, in the reference model. The impact of T_r on the performance of the proposed method will be discussed in Section 6.3.3.

The calculated structural distortion value with Equation 6.2.7 does not have the

same scale as the modulated curvature variation calculated with Equation 6.2.6. To integrate the two values together, we need to scale the two values so that they have similar ranges. According to the experimental results, the structural distortion values are in the range of $[0, 1000]$ for most of the test models. The modulated mean curvature variation is in the range of $[0, 1]$ according to Equation 6.2.5. Therefore we divide the value of structural distortion by 1000 before integrating the two values together. By integrating the mean curvature difference and the structural distortion, a final distortion score D is derived as follows.

$$D(M^r, M^d) = (1 - \alpha) \cdot D_{diff}(M^r, M^d) + \alpha \cdot D_{struct}(M^r, M^d) \quad (6.2.8)$$

where α is a weighting parameter. A larger α gives more emphasis on the structural distortion, and the impact of α on the performance of the proposed method will also be discussed in Section 6.3.3.

By combining the structural distortion with the mean curvature difference, we address the problem of underestimation of smoothing distortion. Figure 6.10(b) illustrates scatter plots of MOSs versus predicted scores by the proposed method based on both mean curvature difference and structural distortion for the models in Figure 6.9(b), (c) and (d), and we can see that the predicted scores exhibit the same rank as the MOSs for the models.

6.3 Experimental Results and Discussions

In this section, we test the performance of the proposed quality assessment method on three publicly available databases: the LIRIS/EPFL general-purpose database [66],

the LIRIS masking database [63], and the UWB compression database [131]. We also compare the performance of the proposed method with relevant existing quality assessment methods including the state-of-the-art ones, in terms of accuracy and robustness. These methods include the Hausdorff distance (HD), Root mean square error (RMS), 3DWPM1 [18], 3DWPM2 [18], MSDM2 [64], DAME [131], FMPD [136], and TPDM [128]. Finally, we discuss the impact of important parameters on the performance of the proposed method.

6.3.1 Databases

Here we give a brief introduction of the three databases used in the experiments. The LIRIS/EPFL general-purpose database [66] includes 4 reference models, and 84 distorted models (21 distorted models for each reference model). Two types of distortions are included in the database: random noise and smoothing distortion, to simulate various artifacts from geometry processing operations, such as compression and simplification. 12 observers are employed in the MOS scoring for the LIRIS/EPFL General-Purpose database. A score between 0 (best) and 10 (worst) is given to each distorted model by observers, and a normalized MOS is computed for each model by averaging the scores given by the observers. The LIRIS masking database [63] includes 4 reference models, and 24 distorted models (6 distorted models for each reference model). The distorted models in the database are created by adding random noise with different strength in rough or smooth regions of the reference models. The purpose of the database is to test the visual masking effect in quality assessment of 3D meshes. 11 observers are employed in the MOS scoring for the LIRIS Masking

database. A normalized score between 0 (worst) and 4 (best) is assigned to each distorted model. The UWB compression database [131] includes 5 reference models, and 63 distorted models (12 or 13 distorted models for each reference model). Thirteen types of artificial distortions related with compression are applied on the reference models. 69 observers are employed in the MOS scoring for the UWB compression database. Observers sorted the distorted models for each reference model according to the perceived quality, and a score between 0 (reference level) and 1 (worst) is computed based on the sorting results. Note that there is no correlation between the scores given to distorted models which correspond to different reference models, due to the procedure of the subjective tests. For example, a distorted model of Bunny with score “1” does not have similar visual quality as a distorted model of Jessi with score “1”. Therefore, the performance of all the test methods on the UWB compression database is derived by averaging the performance on the 5 model sets. The processing of scores in UWB compression database is employed in existing quality metrics [128, 129, 131].

6.3.2 Accuracy and Robustness Evaluation

Figure 6.12 illustrates the scatter plots of the proposed quality assessment method on these three databases. We show the scatter plots of individual 3D models of the UWB compression database in Figure 6.12(c)-(g) due to the lack of inter-object correlation between models corresponding to different reference models. The psychometric function in each plot maps the computed distortion scores to the MOSs. It has been employed by existing quality assessment methods [73, 128, 142] to remove

Figure 6.12: Scatter plots of MOSs versus the predicted scores from the proposed model for the meshes in the three databases. (a) LIRIS/EPFL general-purpose database. (b) LIRIS masking database. (c) Bunny model, (d) James model, (e) Jessi model, (f) Nissan model, (g) Helix model in the UWB compression database.

Table 6.2: Correlation coefficients r_p and r_s (%) of different objective metrics on LIRIS/EPFL general-purpose database.

	Armadillo		Dinosaur		Venus		RockerArm		All models	
	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s
HD	54.9	69.5	47.5	30.9	8.9	1.6	23.4	18.1	11.4	13.8
RMS	56.7	62.7	0.0	0.3	87.9	90.1	17.3	7.3	28.1	26.8
3DWPM1	59.7	65.8	59.7	62.7	68.3	71.6	72.9	87.5	61.9	69.3
3DWPM2	65.6	74.1	44.6	52.4	40.5	34.8	54.7	37.8	49.6	49.0
MSDM2	72.8	81.6	73.5	85.9	76.5	89.3	76.1	89.6	66.2	80.4
DAME	76.3	60.3	88.9	92.8	83.9	91.0	80.1	85.0	75.2	76.6
FMPD	83.2	75.4	88.9	89.6	83.9	87.5	84.7	88.8	83.5	81.9
TPDM	78.8	84.5	89.0	92.2	91.0	90.6	91.4	92.2	86.2	89.6
Proposed	90.5	85.8	90.1	91.4	91.5	89.4	92.1	90.5	88.0	87.0

any nonlinearity from the subjective viewing tests. The function employed in this work has the following form:

$$f(D) = \nu_1 \cdot \left(\frac{1}{2} - \frac{1}{1 + \exp(\nu_2 \cdot (D - \nu_3))} \right) + \nu_4 \cdot D + \nu_5 \quad (6.3.1)$$

where $\nu_1, \nu_2, \nu_3, \nu_4$, and ν_5 are parameters to be fitted by minimizing the sum of squared differences between the mapped distortion and the MOSs. As can be observed from the scatter plots, the proposed method performs well on these three databases. The computed distortion scores exhibit the similar order as the MOSs in most cases.

To compare the performance of the quality assessment methods, two criteria are used. The Pearson linear correlation coefficient (r_p) is employed to measure the prediction accuracy, and the Spearman rank-order correlation coefficient (r_s) is employed to measure the prediction monotonicity. The Spearman rank-order correlation coefficient depends only on the rank of the models; if the rank of objective scores is similar as the rank of MOSs, a high value will be computed, regardless of the distance be-

Table 6.3: Correlation coefficients r_p and r_s (%) of different objective metrics on LIRIS masking database.

	Armadillo		Dinosaur		Bimba		LionVase		All models	
	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s
HD	61.4	48.6	55.8	48.6	27.4	25.7	50.1	71.4	20.2	26.6
RMS	66.8	65.7	70.9	71.4	46.7	71.4	48.8	71.4	41.2	48.8
3DWPM1	64.6	58.0	67.3	66.7	29.0	20.0	31.1	20.0	31.9	29.4
3DWPM2	61.6	48.6	70.8	71.4	37.9	37.1	46.9	38.3	42.7	37.4
MSDM2	65.8	88.6	91.5	100	93.7	100	87.5	94.3	76.2	89.6
DAME	96.0	94.3	89.4	82.9	88.0	97.7	99.5	100	58.6	68.1
FMPD	94.2	88.6	96.9	94.3	98.9	100	93.5	94.3	80.8	80.2
TPDM	91.4	88.6	97.1	100	97.2	100	88.4	82.9	88.6	90.0
Proposed	86.4	88.6	97.5	100	98.5	100	97.8	94.3	92.3	91.6

tween the objective score and the corresponding MOS. For both criteria, a higher value indicates better prediction performance.

The values of r_p and r_s from the objective metrics on the three databases are listed in Table 6.2 to Table 6.4, where the two highest correlation coefficients have been highlighted in boldface. The proposed method has the highest overall r_p and r_s values on the LIRIS masking database (Table 6.3). The LIRIS masking database is designed to evaluate the visual masking effect in quality assessment of 3D meshes. The good performance of the proposed method in the database confirms the effectiveness of the proposed visual masking modulation component in this work. The proposed method has comparable performance to TPDM [128] on the LIRIS/EPFL general-purpose database (Table 6.2), and to DAME [131] on the UWB compression database (Table 6.4). In summary, the proposed method yields consistent results across the three databases. The TPDM [128] performs the best or close to the best on the LIRIS/EPFL general-purpose database and the LIRIS masking database,

Table 6.4: Correlation coefficients r_p and r_s (%) of different objective metrics on UWB compression database.

	Bunny		James		Jessi		Nisaan		Helix		All models	
	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s
HD	34.1	52.2	16.8	6.8	23.6	12.5	14.4	23.6	45.1	46.4	10.6	28.3
RMS	34.2	20.9	14.0	10.8	0.0	14.8	17.8	29.7	46.9	44.6	22.0	24.1
3DWPM1	94.7	93.4	77.3	72.3	87.2	89.5	63.6	59.3	98.0	95.2	84.1	81.9
3DWPM2	96.0	91.2	76.9	65.3	86.9	85.9	56.3	67.6	95.5	94.3	82.3	80.9
MSDM2	97.4	90.1	84.3	63.1	82.6	69.2	84.4	73.1	98.1	94.7	89.3	78.0
DAME	96.8	93.4	84.4	70.5	95.7	93.4	93.9	75.3	96.6	95.2	93.5	85.6
FMPD	94.2	89.6	95.3	91.2	63.3	60.0	92.4	77.5	98.4	90.8	88.8	81.8
TPDM	95.1	96.5	90.8	73.6	85.8	75.8	82.7	73.4	98.7	95.0	91.5	82.9
Proposed	97.6	92.3	96.4	94.3	94.1	74.9	93.9	64.3	98.0	95.2	96.0	84.2

but is slightly worse than the best two methods on the UWB compression database. The DAME [131] has the highest r_s value on the UWB compression database, but is poor on the LIRIS/EPFL general-purpose database and the LIRIS masking database. Overall, the proposed method gives more consistent performance across the three databases.

6.3.3 Impact of Parameters

There are four parameters in the proposed method. The first parameter is σ in Equation 6.2.4, which controls the shape of the masking modulation function. Figure 6.13(a)-(c) plot r_p as a function of σ on the three databases. We use only r_p since r_s leads to similar results. When σ is small (e.g. the curve corresponding to $\sigma = 20$ in Figure 6.13(d)), the masking coefficient decreases rapidly from 1 to 0 as the mean curvature increases, and differences in curvatures for vertices with high curvature val-

Figure 6.13: Plot of r_p as a function of σ for (a) LIRIS/EPFL general-purpose database, (b) LIRIS masking database, (c) UWB compression database. (d) Shape of the masking modulation function with different σ .

ues are masked to 0. The aggressive masking results in a low r_p value. As σ increases, r_p also increases. However, with further increase of σ value, r_p decreases. A large σ (e.g. the curve corresponding to $\sigma = 120$ in Figure 6.13(d)) leads to weak masking of vertices with high curvature values. In an extreme case, when σ approaches positive infinity, the result is the same as the case without visual masking modulation. In this work, we set $\sigma = 45$ empirically.

The second parameter, β in Equation 6.2.5, controls the shape of the saturation function. Figure 6.14(a) plots the correlation coefficient r_p as a function of β on the LIRIS masking database as an example. Figure 6.14(b) illustrates the shape of

Figure 6.14: Impact of β on the performance. (a) Plot of r_p as a function of β on the LIRIS masking database. (b) Shape of the saturation modulation function with different β .

the saturation modulation function with different β . In an extreme case, when β approaches 0, the curvature difference values are all mapped to value “1”; when β approaches positive infinity, the curvature variation values are all mapped to value “0”. Thus as shown in Figure 6.14(a), with the increase of β , the performance increases first but decreases later. We choose $\beta = 30$ in this work, and it gives a stable results for all the test models.

The third parameter, T_r in Equation 6.2.7, is an important parameter in the structural distortion computation. If T_r is too small, non-edge vertices are included in the structural distortion; while if T_r is too large, some edge vertices are ignored. Figure 6.15 plots the correlation coefficient r_p as a function of T_r for the models created by smoothing operation in the LIRIS/EPFL general database. We choose this database since it contain models distorted by smoothing operation. The correlation coefficient r_p first increases with the increase of T_r , since more non-edge vertices are excluded in the computation. However, further increase in T_r leads to decrease of r_p ,

Figure 6.15: Plot of r_p as a function of T_r for the models distorted by smoothing operation in the LIRIS/EPFL general-purpose database.

since some edge vertices are not considered in the structural distortion computation. In this work, we set $T_r = 70$.

The last parameter, α in Equation 6.2.8, determines the weight of the structural distortion in the overall distortion. Figure 6.16 plots the correlation coefficient r_p as a function of α for the models in the LIRIS/EPFL general database and the LIRIS masking database. We choose the two databases to study the impact of α because the two databases includes sufficient number of distorted models for each distortion type (for each reference model in the UWB compression database, there is only one distorted model for each distortion type). As shown in the plot (Figure 6.16), the structural distortion is less effective in measuring the noise distortion: with the increase of weight of the structural distortion, the performance on the LIRIS masking database decreases. With the increase of weight of the structural distortion, the performance on the LIRIS/EPFL general database first increases slightly, and then decreases. The reason is that the database includes two distortion types: noise distortion and smoothing distortion. Although structural distortion is effective in measuring the smoothing

Figure 6.16: Plot of r_p as a function of α for the models in the LIRIS/EPFL general-purpose database and the LIRIS masking database.

distortion, it is not so effective in measuring the noise distortion. Considering that our purpose is to design a general-purpose quality assessment method, which should be useful for general types of distortion instead of one specific distortion type such as smoothing distortion, we choose $\alpha = 0.13$ in this work, and all the experimental results presented in this work are generated with the same set of parameters.

6.4 Concluding Remarks

We have designed and implemented a new method to assess the perceptual quality of 3D triangle meshes. Given a distorted 3D model and a reference one, we compute mean curvature of vertex of both models and compare the difference in mean curvature between each pair of corresponding vertices. The mean curvature variation is modulated with a visual masking module and a saturation module. With the visual masking module, computed curvature differences in high-curvature vertices are masked by the curvature values. With the saturation module, high-level curvature

differences are decreased to account for the decrease in sensitivity of the HVS at high distortion levels. The structural distortion is computed based on the difference in maximum curvature between edge vertices of the two models. A distortion score is derived as the combination of the mean curvature variation and the structural distortion. Compared with relevant existing quality assessment methods for 3D triangle meshes, the proposed method yields more consistent results with human perception on three publicly available databases.

Chapter 7

Summary and Future Work

7.1 Summary

To improve the effectiveness and efficiency of computer graphic rendering, this thesis has presented a number of approaches which exploit the perceptual properties of the HVS in the following aspects: incorporation of visual saliency in perceptual rendering, incorporation of visual masking in perceptual rendering, and user-oriented quality evaluation of 3D meshes for possible use in practice.

The major technical contributions of this project have been highlighted in Section 1.3, and in this section, we will give a summary of the research that has been performed in this project, with links back into the preceding chapters.

7.1.1 Incorporation of Visual Saliency in Perceptual Rendering

Existing visual saliency based perceptual rendering methods try to accelerate rendering process by distributing less computational resources to pixels of lower visual saliency. These methods use the saliency level of pixels as the only guide for allocation of rendering computation. As a result, rendering artifacts will be introduced to pixels with high rendering complexity. The artifacts may turn the original non-salient regions into salient ones, and degrade the overall quality of rendered image. The problem is illustrated with examples in Section 3.1. To address this problem in the context of ray-tracing based rendering algorithms, in Section 3.2.2 we employ an error metric to estimate the rendering complexity of pixels: at a low computation, pixels exhibiting larger error values need more computation to achieve an acceptable quality than other pixels. We combine the complexity estimation result and the visual saliency detection result into a priority map, and use the priority value as guide for allocation of computation. We propose an adaptive sampling scheme for the allocation of computation and the scheme is illustrated in Figure 3.3.

The results in Section 3.3.1 have firstly confirmed that the proposed scheme overcomes the shortcomings in the existing methods based upon visual saliency only. The subsequent subjective experiments in Section 3.3.2 have shown that under an equal-quality comparison, the proposed perceptual rendering scheme decreases the computational complexity, compared with a recent non-perceptual rendering scheme.

We also explore a problem with saliency-based perceptual rendering: accurate boundaries of salient objects cannot be detected by the existing saliency detection

methods; the inaccurate saliency detection results cause the change in quality within a salient object in a rendered image and degrade the perceived quality of the image. To improve the accuracy of saliency detection as well as to address the problem in saliency-based perceptual rendering, we propose an enhanced visual saliency detection method for rendered graphics in Chapter 4. Existing graphics saliency detection methods compute pixel-level contrast, and cannot detect accurate boundaries of salient objects. To address this problem, the proposed visual saliency detection method computes object-level contrast by exploiting the accurate segmentation information extracted from 3D models. Sections 4.2.1 and 4.2.2 describe how to extract the color feature for each segmented object, and Section 4.2.3 describes the computation of object-level contrast between each object and its surrounding objects. The final saliency map is generated by integrating the object-level contrast of each segmented object.

Experimental results in Section 4.3 demonstrate that the enhanced visual saliency detection method yields more accurate saliency detection results than the relevant existing saliency detection methods. We incorporate the proposed visual saliency detection in the perceptually based adaptive sampling scheme described in Chapter 3, and compare the performance with the relevant existing saliency detection methods in the same adaptive sampling scheme. The experimental results in Section 4.4 demonstrate that with the proposed saliency detection method, the problem of quality inconsistency within a single salient object is addressed in saliency-based perceptual rendering.

7.1.2 Incorporation of Visual Masking in Perceptual Rendering

Besides the saliency-based perceptual rendering, we explore the incorporation of visual masking in perceptual rendering in Chapter 5. According to the existing studies [138], visibility of visual signals (e.g. noise) decreases due to the presence of a mask signal which is unknown to human eyes. In the context of rendering, the property indicates that rendering artifacts are less visible in irregular image regions. To exploit the perceptual property in rendering, we derive an irregularity measure in Section 5.2.2. The underlying idea is that: for regular regions, the differences between center pixels and their neighboring pixels are small; for irregular regions, the differences between center pixels and their neighboring pixels are large. For a pixel in the original image, a similarity-weighted sum of the neighboring pixels is computed, and the irregularity of the pixel is calculated as the difference between the pixel value and the weighted sum of the neighboring pixels. Examples presented in Section 5.2.2 demonstrate that irregular regions are well distinguished from irregular regions with the proposed measure.

We incorporate the irregularity measure in perceptual rendering, and the scheme is described in Section 5.2.3. We also compare the performance with the relevant existing visual masking methods in the same perceptual rendering scheme. Experimental results in Section 5.3.1 demonstrate that, with the same computational resources, the proposed method generates images of better perceived quality than the relevant existing visual masking schemes. We conduct an equal-quality comparison between the proposed perceptual rendering scheme and a recent non-perceptual rendering scheme.

Results in Section 5.3.2 demonstrate that incorporating the irregularity measure in rendering accelerates rendering significantly.

7.1.3 Quality Evaluation for 3D Meshes

An objective quality evaluation method for 3D meshes is described in Chapter 6. The proposed method relies on both the difference in mean curvature and the structural distortion. The pipeline of the proposed method is described in Section 6.2. Given a reference mesh and a distorted mesh, the proposed method calculates the difference in curvature values between each pair of vertices in the two meshes. The curvature difference between each pair of vertices is modulated by two perceptual modules: the visual masking module and the saturation module.

Section 6.2.3 describes the proposed visual masking component, with which the curvature difference is modulated by an exponential function. Curvature differences in rough regions are decreased to account for the decrease in sensitivity of the HVS to distortion in rough regions. Section 6.2.4 describes the proposed saturation component. High-level curvature differences are decreased to account for the fact that the sensitivity of the HVS to difference decreases at high difference levels. Additionally, Section 6.2.5 describes the method to compute the structure dissimilarity between the two meshes. Finally, a distortion score is derived by pooling the modulated curvature difference and the structural distortion. Experimental results in Section 6.3 demonstrate that the proposed method yields more consistent results on three publicly available databases, compared with the relevant existing quality assessment methods for 3D meshes.

7.2 Limitations

One limitation of the work documented in the thesis is the small sample size of human evaluators employed in the subjective viewing tests. In the subjective tests described in Chapters 3 and 5, forty-eight human observers were employed to evaluate the performance of the proposed perceptual rendering methods. In the subjective tests described in Chapter 4, nine participants were employed in the task of labeling the ground truth saliency maps. In order to get more convincing conclusions, a larger sample size of human evaluators is needed in the future work. Besides the sample size of human evaluators, the choice of test scenes is also a limitation of the work. The performance of the test methods is influenced by the particular choice of scenes. For example, the proposed perceptual rendering method with irregularity measure cannot accelerate rendering for scenes which do not contain irregular texture. We try to make the test scenes to cover various situations, including indoor scenes and more complex outdoor scenes. However, the lack of a test database of 3D scenes is a major limitation of the proposed work. In the proposed perceptual rendering methods, we incorporate the perceptual models in path tracing with a specific sampling method, i.e. low discrepancy sampling. Another limitation of the proposed work is that the possibilities of modulating other variables, besides the sample per pixel, have not been explored.

In the proposed perceptual rendering methods, we studied the issue of rendering acceleration of static images. In practical applications, however, rendering a single image is of minimal interest and rendering a sequence is more significant. Thus it is important to extend the proposed techniques to the process of rendering a sequence.

In the case of rendering a sequence, a single image is displayed only for a short period of time, and it is more likely that observers focus on salient regions and ignore other parts. It is therefore expected that the benefit brought by incorporation of visual saliency in rendering a sequence would be more significant than rendering a single image.

To incorporate a visual saliency detection model in the rendering process of a sequence, two potential problems need to be solved. First, motion is an important saliency cue, e.g. a moving object in front of a static background attracts more attention than other parts in a sequence, and motion saliency needs to be considered in the saliency detection. Second, temporal flicker in saliency detection results between successive frames should be eliminated. If the saliency detection result changes fast from one frame to the next frame, the flicker in rendering quality between two successive frames will result in unpleasant viewing experience. Thus temporal filtering should be incorporated in the saliency detection. To extend the proposed visual masking based perceptual rendering technique to a sequence, the flicker problem also needs to be tackled by applying temporal filtering to the visual masks of successive frames.

7.3 Future Work

In the previous chapters, we tackle some important problems in perceptual graphic rendering and quality evaluation. However, there are still problems unsolved in perceptual rendering and quality evaluation. In this section, we discuss these possible directions for future work.

7.3.1 Integration of More Visual Perceptual Characteristics

One possible direction in perceptual graphic rendering, in our opinion, is the integration of more visual perceptual characteristics. In general, the visual perceptual characteristics can be classified into two categories: low-level visual perception and high-level visual perception [57]. The study of low-level visual perception is concerned with the HVS component that is driven by sensory input to detect lines, shapes, colors, textures, and depth, while high-level visual perception is concerned with the HVS component which implements mental imagery and object identification [57].

Compared with high-level visual perception, such as the the visual saliency, low-level visual perception has been well understood and modeled, and its related psychological mechanisms can be easily modeled computationally. For example, the human eye's non-linear response to luminance can be modeled with the threshold-vs-intensity (TVI) function described in [30], and the HVS's spatiotemporal contrast sensitivity can be modeled with the CSF [101]. The low-level vision perception is easy to be modeled and not influenced much by user experience, such as preference, expectation and knowledge background. However, the low-level vision perception based models are sometimes too conservative when applied in rendering application since the perceptual quality can be preserved even when the rendered image is visually different from a reference [106]. Therefore high-level visual perception based models have been proposed and applied in rendering applications such as the visual saliency detection models and the VEP [106] as well as their refined version [58]. High-level visual perception based models can achieve higher performance improvement compared with their low-level counterparts. But the major problems with the high-level visual per-

ception based models are: 1) most of the high-level visual processing mechanisms cannot be easily modeled since only a small parts of the high-level visual processing have been well understood so far; 2) high-level visual processing is usually influenced by user experience, which makes a generalized model more difficult.

Considering that both the low-level and high-level visual perception based models have their advantages and disadvantages, integrating the two components may offer a better solution. Yee [144] made an early attempt to integrate the low-level visual perception and the visual saliency in his perceptual model, the visual saliency was considered in determination of spatiotemporal contrast sensitivity of the HVS. Some other existing work on the integration of the visual saliency and low-level visual perception include the derivation of attention-based quality metrics for natural images. But whether the simple combination between attention map and quality map is appropriate for the integration still remains open. To integrate the visual saliency with low-level visual perception for perceptual rendering, we are to firstly combine the visual saliency detection model with the contrast masking mechanism, and then based on the result analysis, we will try to integrate other aspects of the low-level visual perception into the visual saliency detection model to achieve better perceptual rendering performance.

Different high-level visual perceptual characteristics can also be integrated. For example, it is possible to integrate the visual saliency described in Chapters 3 and 4 and the visual masking of irregular structure described in Chapter 5. Visual saliency has been used for reduction of computational cost in regions where the HVS does not pay attention to, and visual masking has been used for computational cost reduction in regions where the sensitivity of the HVS to distortion is low. Considering that

non-salient regions and irregular regions are not relevant in general (the regions may overlap in some cases), we will integrate visual saliency and irregularity measure in perceptual rendering for more significant computational saving. Task [122] and memory related clues [7, 148] can also be integrated with visual saliency and irregularity measure in perceptual rendering.

7.3.2 Quality Evaluation of 3D Models and Rendered Graphics

In Chapter 6, we discuss a perceptual quality evaluation method for 3D models. One possible future direction is to extend the quality evaluation of non-textured 3D models to textured 3D models. Although most 3D models used in realistic graphic rendering are textured, the 3D models used in existing quality evaluation databases [63, 66, 131] are all non-textured. Some existing quality metrics have been proposed for textured 3D model [17, 127], in which both geometry and texture information is used. The resolutions of geometry and texture are considered in the metrics [17, 127]. However, the properties of the HVS, e.g. the texture masking effect [31, 104], have not been considered in the quality metrics for textured 3D models. To design quality metrics which correlate better with human perception, we will study perceptual properties related with texture and color information, and incorporate these properties in the quality metric. To measure the performance of newly proposed metrics for textured 3D models, a subjective database of textured 3D models also needs to be built.

Quality metrics of 3D models can be employed in graphic rendering, e.g. in a LoD rendering scheme [81]. In a LoD rendering scheme, objects are represented with

different resolutions: low-resolution objects are used in rendering to save rendering cost [44]. LoD scheme is a trade-off between performance and complexity. With the objective quality metric, the proper resolution can be selected automatically according to the objective distortion level.

Possible future directions also include studying the quality evaluation of rendered graphics and incorporating more comprehensive quality metrics in perceptual rendering. Quality metrics for rendered graphics are widely used in perceptual rendering [9, 10, 24, 29, 31, 80, 88, 89, 91, 107]. The quality metrics used in existing work are predominantly full-reference based. Given a reference image and a test image, a visual metric predicts the visible error between the two images. Based on the prediction results, computational resources are distributed adaptively to regions with high levels of visible error. The major limitation of these perceptual rendering algorithms is the lack of a perfect reference image. In general, a perfect reference image without rendering artifact is not available in perceptual rendering process. Approximated images are generated by direct illumination [107], approximation in the frequency domain [10], intermediate rendering results [88], and so on. However, prediction errors are introduced, as a result of using the approximated image, and the quality of rendered images may be affected. To improve the prediction accuracy of visual metrics in perceptual rendering, a no-reference quality metric for rendered graphics needs to be designed. Herzog *et al.* [42] proposed a no-reference quality metric for rendered images by training a framework with typical rendering artifacts. The metric is designed for three types of rendering distortions: VPL clamping, glossy VPL noise and shadow map aliasing. To design a general-purpose no-reference quality metric, we will study how to select appropriate features related to quality of rendered graphics,

and use training methods via machine learning to generate a quality predictor based on the extracted features.

Bibliography

- [1] P. L. C. A. Ninassi, O. Le Meur and D. Barba. On the performance of human visual system based image quality assessment metric using wavelet domain. *SPIE proceedings Human Vision and Electronic Imaging XIII*, 6806, 2008.
- [2] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, and M. Desbrun. Anisotropic polygonal remeshing. *ACM Transactions on Graphics*, 22(3):485–493, 2003.
- [3] J. R. Anderson. *Cognitive psychology and its implications*. WH Freeman/Times Books/Henry Holt & Co, 1990.
- [4] O. Anson, V. Sundstedt, D. Gutierrez, and A. Chalmers. Efficient selective rendering of participating media. In *Proceedings of the 3rd symposium on Applied perception in graphics and visualization*, pages 135–142, 2006.
- [5] M. Aranha, K. Debattista, A. Chalmers, and S. Hill. Perceived Rendering Thresholds for High-Fidelity Graphics on Small Screen Devices. *Theory and Practice of Computer Graphics*, pages 133–140, 2006.
- [6] D. Bartz, D. Cunningham, J. Fischer, and C. Wallraven. The role of perception for computer graphics. *Eurographics state-of-the-art-reports*, pages 65–86, 2008.

- [7] A. Bennett, M. Coxon, and K. Mania. The effect of stereo and context on memory and awareness states in immersive virtual environments. In *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, pages 135–140, 2010.
- [8] Z. Bian, S.-M. Hu, and R. R. Martin. Evaluation for small visual difference between conforming meshes on strain field. *Journal of Computer Science and Technology*, 24(1):65–75, 2009.
- [9] M. R. Bolin and G. W. Meyer. A frequency based ray tracer. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 409–418. ACM, 1995.
- [10] M. R. Bolin and G. W. Meyer. A perceptually based adaptive sampling algorithm. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 299–309. ACM, 1998.
- [11] V. Bruce, P. R. Green, and M. A. Georgeson. *Visual perception: Physiology, psychology, & ecology*. Psychology Press, 2003.
- [12] K. Cater, A. Chalmers, and P. Ledda. Selective quality rendering by exploiting human inattentional blindness: looking but not seeing. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 17–24, 2002.
- [13] K. Cater, A. Chalmers, and G. Ward. Detail to attention: exploiting visual tasks for selective rendering. In *Proceedings of the 14th Eurographics workshop on Rendering*, pages 270–280, 2003.

- [14] A. Chalmers, K. Debattista, G. Mastoropoulou, and L. Paulo dos Santos. There-reality: selective rendering in high fidelity virtual environments. *The International Journal of Virtual Reality*, 6(1):1–10, 2007.
- [15] A. Chalmers, K. Debattista, and B. Ramic-Brkic. Towards high-fidelity multi-sensory virtual environments. *The Visual Computer*, 25(12):1101–1108, 2009.
- [16] I. Cheng and A. Basu. Perceptually optimized 3-d transmission over wireless networks. *IEEE Transactions on Multimedia*, 9(2):386–396, 2007.
- [17] I. Cheng and P. Boulanger. A visual quality prediction model for 3d texture. In *Proceedings EUROGRAPHICS*, pages 101–104, 2005.
- [18] M. Corsini, E. Gelasca, T. Ebrahimi, and M. Barni. Watermarked 3-d mesh quality assessment. *IEEE Transactions on Multimedia*, 9(2):247–256, Feb 2007.
- [19] M. Corsini, M.-C. Larabi, G. Lavoué, O. Petřík, L. Váša, and K. Wang. Perceptual metrics for static and dynamic triangle meshes. *Computer Graphics Forum*, 32(1):101–125, 2013.
- [20] S. Daly. *Digital images and human vision*. MIT Press, Cambridge, MA, USA, 1993.
- [21] Y. Deng, B. S. Manjunath, C. Kenney, M. S. Moore, and H. Shin. An efficient color representation for image retrieval. *IEEE Transactions on Image Processing*, 10(1):140–147, 2001.
- [22] M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes. *Computer Graphics Forum*, 21(3):209–218, 2002.

- [23] L. Dong, W. Lin, Y. Fang, S. Wu, and H. S. Seah. Saliency detection in computer rendered images based on object-level contrast. *Journal of Visual Communication and Image Representation*, 25(3):525 – 533, 2014.
- [24] R. Dumont, F. Pellacini, and J. A. Ferwerda. Perceptually-driven decision theory for interactive realistic rendering. *ACM Transactions on Graphics (TOG)*, 22(2):152–181, 2003.
- [25] K. Egan, F. Hecht, F. Durand, and R. Ramamoorthi. Frequency analysis and sheared filtering for shadow light fields of complex occluders. *ACM Transactions on Graphics*, 30(2):9:1–9:13, Apr. 2011.
- [26] K. Egan, Y.-T. Tseng, N. Holzschuch, F. Durand, and R. Ramamoorthi. Frequency analysis and sheared reconstruction for rendering motion blur. *ACM Transactions on Graphics*, 28(3):93:1–93:13, July 2009.
- [27] W. B. Engel S, Zhang X. Colour tuning in human visual cortex measured with functional magnetic resonance imaging. *Nature*, 388(6637):68–71, July 1997.
- [28] Y. Fang, W. Lin, B.-S. Lee, C.-T. Lau, Z. Chen, and C.-W. Lin. Bottom-up saliency detection model based on human visual sensitivity and amplitude spectrum. *IEEE Transactions on Multimedia*, 14(1):187–198, Feb. 2012.
- [29] J.-P. Farrugia and B. Proche. A progressive rendering algorithm using an adaptive perceptually based image metric. *Computer Graphics Forum*, 23:605–614, 2004.
- [30] J. A. Ferwerda, S. N. Pattanaik, P. Shirley, and D. P. Greenberg. A model

- of visual adaptation for realistic image synthesis. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 249–258. ACM, 1996.
- [31] J. A. Ferwerda, P. Shirley, S. N. Pattanaik, and D. P. Greenberg. A model of visual masking for computer graphics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 143–152, 1997.
- [32] V. Gopalakrishnan, Y. Hu, and D. Rajan. Salient region detection by modeling distributions of color and orientation. *IEEE Transactions on Multimedia*, 11(5):892–905, 2009.
- [33] D. P. Greenberg, K. E. Torrance, P. Shirley, J. Arvo, E. Lafortune, J. A. Ferwerda, B. Walter, B. Trumbore, S. Pattanaik, and S.-C. Foo. A framework for realistic image synthesis. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 477–494. ACM, 1997.
- [34] C. Guo, Q. Ma, and L. Zhang. Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [35] Y. Guo, F. Liu, J. Shi, Z.-H. Zhou, and M. Gleicher. Image retargeting using mesh parametrization. *IEEE Transactions on Multimedia*, 11(5):856–867, Aug. 2009.
- [36] J. Haber, K. Myszkowski, H. Yamauchi, and H.-P. Seidel. Perceptually guided corrective splatting. 20(3):142–153, 2001.

- [37] T. Hachisuka, W. Jarosz, R. P. Weistroffer, K. Dale, G. Humphreys, M. Zwicker, and H. W. Jensen. Multidimensional adaptive sampling and reconstruction for ray tracing. *ACM Transactions on Graphics*, 27(3):33:1–33:10, Aug. 2008.
- [38] T. Hachisuka and H. W. Jensen. Parallel progressive photon mapping on gpus. In *ACM SIGGRAPH ASIA 2010 Sketches*, page 54. ACM, 2010.
- [39] C. Harvey, T. Bashford-Rogers, K. Debattista, and A. Chalmers. Visual saliency for smell impulses and application to selective rendering. *Theory and practice of computer graphics 2011: Eurographics UK Chapter proceedings*, pages 73–80, 2011.
- [40] J. Hasic, A. Chalmers, and E. Sikudova. Perceptually guided high-fidelity rendering exploiting movement bias in visual attention. *ACM Transactions on Applied Perception*, 8(1):6, 2010.
- [41] D. Hearn and M. P. Baker. *Computer Graphics with Open GL, 3/E*. Prentice Hall, 2004.
- [42] R. Herzog, M. Čadík, T. O. Aydčín, K. I. Kim, K. Myszkowski, and H.-P. Seidel. Norm: No-reference image quality metric for realistic image synthesis. 31(2pt3):545–554, 2012.
- [43] S. Hillaire, A. Lecuyer, T. Regia-Corte, R. Cozot, J. Royan, and G. Breton. Design and application of real-time visual attention model for the exploration of 3d virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 18(3):356–368, 2012.

- [44] H. Hoppe. Smooth view-dependent level-of-detail control and its application to terrain rendering. In *Proceedings IEEE Visualization*, pages 35–42, 1998.
- [45] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.
- [46] V. Hulusic, C. Harvey, K. Debattista, N. Tsingos, S. Walker, D. Howard, and A. Chalmers. Acoustic rendering and auditory–visual cross-modal perception and interaction. 31(1):102–131, 2012.
- [47] L. Itti and C. Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40(10-12):1489–1506, 2000.
- [48] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, Nov. 1998.
- [49] D. J. Selective attention and the organization of visual information. *Journal of Experimental Psychology: General*, 113(4):501–517, 1984.
- [50] H. W. Jensen. *Realistic image synthesis using photon mapping*. AK Peters, Ltd., 2001.
- [51] J. Jimenez, V. Sundstedt, and D. Gutierrez. Screen-space perceptual rendering of human skin. *ACM Transactions on Applied Perception (TAP)*, 6(4):23, 2009.
- [52] B. Jin, I. Ihm, B. Chang, C. Park, W. Lee, and S. Jung. Selective and adaptive

- supersampling for real-time ray tracing. In *Proceedings of the Conference on High Performance Graphics*, pages 117–125, 2009.
- [53] J. Jonides. Further towards a model of the mind’s eye’s movemen. *Bulletin of the Psychonomic Society*, 21(4):247–250, 1983.
- [54] J. T. Kajiya. The rendering equation. In *ACM Siggraph Computer Graphics*, volume 20, pages 143–150. ACM, 1986.
- [55] N. K. Kalantari and P. Sen. Removing the noise in Monte Carlo rendering with general image denoising algorithms. *Computer Graphics Forum*, 32(2), 2013.
- [56] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 279–286, 2000.
- [57] S. Kosslyn, R. Flynn, J. Amsterdam, and G. Wang. Components of high-level vision: A cognitive neuroscience analysis and accounts of neurological syndromes* 1. *Elsevier Journal of Cognition*, 34(3):203–277, 1990.
- [58] J. Křivánek, J. Ferwerda, and K. Bala. Effects of global illumination approximations on material appearance. *ACM Transactions on Graphics*, 29(4):112:1–112:10, 2010.
- [59] J. Křivánek and P. Gautron. Practical global illumination with irradiance caching. *Synthesis lectures on computer graphics and animation*, 4(1):1–148, 2009.

- [60] E. P. Lafortune. *Mathematical models and Monte Carlo algorithms for physically based rendering*. PhD thesis, Katholieke Universiteit Leuven, 1996.
- [61] E. P. Lafortune and Y. D. Willems. Rendering participating media with bidirectional path tracing. In *Proceedings Rendering Techniques*, pages 91–100. Springer, 1996.
- [62] M. Larkin and C. O’Sullivan. Perception of simplification artifacts for animated characters. In *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*, pages 93–100. ACM, 2011.
- [63] G. Lavoué. A local roughness measure for 3d meshes and its application to visual masking. *ACM Transactions on Applied Perception*, 5(4):21:1–21:23, 2009.
- [64] G. Lavoué. A multiscale metric for 3d mesh visual quality assessment. *Computer Graphics Forum*, 30(5):1427–1437, 2011.
- [65] G. Lavoué and M. Corsini. A comparison of perceptually-based metrics for objective evaluation of geometry processing. *IEEE Transactions on Multimedia*, 12(7):636–649, 2010.
- [66] G. Lavoué, E. D. Gelasca, F. Dupont, A. Baskurt, and T. Ebrahimi. Perceptually driven 3d distance metrics with application to watermarking. In *SPIE Optics+ Photonics*, pages 63120L–63120L, 2006.
- [67] C. H. Lee, A. Varshney, and D. W. Jacobs. Mesh saliency. *ACM Transactions on Graphics*, 24(3):659–666, July 2005.

- [68] S. Lee, G. Jounghyun Kim, and S. Choi. Real-time tracking of visually attended objects in virtual environments and its application to lod. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):6–19, 2009.
- [69] S. Lee, G. J. Kim, and S. Choi. Real-time tracking of visually attended objects in interactive virtual environments. In *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, pages 29–38. ACM, 2007.
- [70] J. Lehtinen, T. Aila, J. Chen, S. Laine, and F. Durand. Temporal light field reconstruction for rendering distribution effects. *ACM Transactions on Graphics*, 30(4):55:1–55:12, Aug. 2011.
- [71] P. Lindstrom. *Model simplification using image and geometry-based metrics*. PhD thesis, Georgia Institute of Technology, 2000.
- [72] P. Lindstrom and G. Turk. Image-driven simplification. *ACM Transactions on Graphics*, 19(3):204–241, 2000.
- [73] A. Liu, W. Lin, and M. Narwaria. Image quality assessment based on gradient similarity. *IEEE Transactions on Image Processing*, 21(4):1500–1512, April 2012.
- [74] A. Liu, W. Lin, M. Paul, C. Deng, and F. Zhang. Just noticeable difference for images with decomposition model for separating edge and textured regions. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(11):1648–1652, 2010.
- [75] C.-H. Lo, C.-H. Chu, K. Debattista, and A. Chalmers. Selective rendering for

- efficient ray traced stereoscopic images. *The Visual Computer*, 26(2):97–107, 2010.
- [76] P. Longhurst, K. Debattista, and A. Chalmers. Snapshot: A rapid technique for driving a selective global illumination renderer. In *Proceedings International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 81–84, 2005.
- [77] P. Longhurst, K. Debattista, and A. Chalmers. A GPU based saliency map for high-fidelity selective rendering. In *Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pages 21–29, 2006.
- [78] P. Longhurst, K. Debattista, R. Gillibrand, and A. Chalmers. Analytic antialiasing for selective high fidelity rendering. In *Proceedings Brazilian Symposium on Computer Graphics and Image Processing*, pages 359–366. IEEE, 2005.
- [79] J. Lubin. *A Visual Discrimination Model for Imaging System Design and Evaluation*, pages 245–283. World Scientific, 1995.
- [80] D. Luebke and B. Hallen. *Perceptually driven simplification for interactive rendering*, pages 223–234. Springer Vienna, 2001.
- [81] D. P. Luebke. A developer’s survey of polygonal simplification algorithms. *IEEE Computer Graphics and Applications*, 21(3):24–35, 2001.
- [82] D. P. Luebke. *Level of detail for 3D graphics*. Morgan Kaufmann, 2003.

- [83] S. P. Maity and M. K. Kundu. Dht domain digital watermarking with low loss in image informations. *AEU - International Journal of Electronics and Communications*, 64(3):243 – 257, 2010.
- [84] B. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, 2001.
- [85] R. Mantiuk, K. J. Kim, A. G. Rempel, and W. Heidrich. Hdr-vdp-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Transactions on Graphics*, 30(4):40:1–40:14, Aug. 2011.
- [86] G. Mastoropoulou, K. Debattista, A. Chalmers, and T. Troscianko. Auditory bias of visual attention for perceptually-guided selective rendering of animations. In *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 363–369. ACM, 2005.
- [87] D. P. Mitchell. Generating antialiased images at low sampling densities. In *ACM SIGGRAPH Computer Graphics*, pages 65–72, 1987.
- [88] K. Myszkowski. The visible differences predictor: Applications to global illumination problems. In *Rendering Techniques 98*, pages 223–236. Springer, 1998.
- [89] K. Myszkowski, P. Rokita, and T. Tawara. Perceptually-informed accelerated rendering of high quality walkthrough sequences. In *Proceedings of the Tenth Eurographics Workshop on Rendering*, pages 5–18, 1999.

- [90] K. Myszkowski, P. Rokita, and T. Tawara. Perception-based fast rendering and antialiasing of walkthrough sequences. *IEEE Transactions on Visualization and Computer Graphics*, 6:360–379, October 2000.
- [91] K. Myszkowski, T. Tawara, H. Akamine, and H.-P. Seidel. Perception-guided global illumination solution for animation rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 221–230. ACM, 2001.
- [92] F. Navarro, S. Castillo, F. J. Serón, and D. Gutierrez. Perceptual considerations for motion blur rendering. *ACM Transactions on Applied Perception*, 8(3):20, 2011.
- [93] A. Ninassi, O. Le Meur, P. Le Callet, and D. Barba. Considering temporal variations of spatial visual distortions in video quality assessment. *IEEE Journal of Selected Topics in Signal Processing*, 3(2):253–265, 2009.
- [94] K. M. O’Craven, P. E. Downing, and N. Kanwisher. fmri evidence for objects as the units of attentional selection. *Nature*, 401(6753):584–587, 1999.
- [95] A. Oliva, A. Torralba, M. Castelhana, and J. Henderson. Top-down control of visual attention in object detection. In *Proceedings International Conference on Image Processing*, volume 1, pages I–253–6 vol.1, Sept 2003.
- [96] C. O’Sullivan, S. Howlett, Y. Morvan, R. McDonnell, and K. O’Conor. Perceptually adaptive graphics. *Eurographics state of the art reports*, 4, 2004.

- [97] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.
- [98] N. Ouerhani and H. Hügli. A model of dynamic visual attention for object tracking in natural image sequences. In *Computational Methods in Neural Modeling*, volume 2686, pages 702–709. 2003.
- [99] R. S. Overbeck, C. Donner, and R. Ramamoorthi. Adaptive wavelet rendering. *ACM Transactions on Graphics*, 28(5):140:1–140:12, Dec. 2009.
- [100] A. Pajot, L. Barthe, M. Paulin, and P. Poulin. Combinatorial bidirectional path-tracing for efficient hybrid cpu/gpu rendering. In *Computer Graphics Forum*, volume 30, pages 315–324. Wiley Online Library, 2011.
- [101] S. N. Pattanaik, J. A. Ferwerda, M. D. Fairchild, and D. P. Greenberg. A multiscale model of adaptation and spatial vision for realistic image display. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 287–298. ACM, 1998.
- [102] M. Pharr and G. Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2004.
- [103] M. Pharr and G. Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2010.
- [104] L. Qu and G. W. Meyer. Perceptually guided polygon reduction. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1015–1029, 2008.

- [105] G. Ramanarayanan, K. Bala, and J. A. Ferwerda. Perception of complex aggregates. 27(3):60, 2008.
- [106] G. Ramanarayanan, J. Ferwerda, B. Walter, and K. Bala. Visual equivalence: Towards a new standard for image fidelity. *ACM Transactions on Graphics*, 26(3), 2007.
- [107] M. Ramasubramanian, S. N. Pattanaik, and D. P. Greenberg. A perceptually based physical error metric for realistic image synthesis. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 73–82. ACM Press/Addison-Wesley Publishing Co., 1999.
- [108] B. Ramic, A. Chalmers, J. Hasic, and S. Rizvic. Selective rendering in a multi-modal environment: Scent and graphics. In *ACM SIGGRAPH Spring Conference on Computer Graphics*, 2006.
- [109] B. Ramic-Brkic, A. Chalmers, A. Sadzak, K. Debattista, and S. Sultanic. Exploring multiple modalities for selective rendering of virtual environments. In *Spring Conference on Computer Graphics*, pages 91–98. ACM, 2013.
- [110] B. E. Rogowitz and H. E. Rushmeier. Are image quality metrics adequate to evaluate the quality of geometric objects? In *Photonics West 2001-Electronic Imaging*, pages 340–348, 2001.
- [111] F. Rousselle, C. Knaus, and M. Zwicker. Adaptive sampling and reconstruction using greedy error minimization. *ACM Transactions on Graphics*, 30(6):159:1–159:12, Dec. 2011.

- [112] F. Rousselle, C. Knaus, and M. Zwicker. Adaptive rendering with non-local means filtering. *ACM Transactions on Graphics*, 31(6):195:1–195:11, 2012.
- [113] A. H. Sadr and S. Ghaemmaghami. Robustness enhancement of content-based watermarks using entropy masking effect. In *Proceedings International Conference on Digital Watermarking*, pages 444–458, 2005.
- [114] B. Scholl. Objects and attention: the state of the art. *Journal of Experimental Psychology: General*, 80(1-2):1–46, 2001.
- [115] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.
- [116] L. Sharan, Z. H. Neo, K. Mitchell, and J. K. Hodgins. Simulated motion blur does not improve player experience in racing game. In *Proceedings of the Motion on Games*, pages 149–154. ACM, 2013.
- [117] P. Shirley, C. Wang, and K. Zimmerman. Monte carlo techniques for direct lighting calculations. *ACM Transactions on Graphics*, 15(1):1–36, 1996.
- [118] P. P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. 21(3):527–536, 2002.
- [119] C. Soler, K. Subr, F. Durand, N. Holzschuch, and F. Sillion. Fourier depth of field. *ACM Transactions on Graphics*, 28(2):18:1–18:12, May 2009.
- [120] O. Sorkine, D. Cohen-Or, and S. Toledo. High-pass quantization for mesh en-

- coding. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 42–51, 2003.
- [121] W. Stokes, J. Ferwerda, B. Walter, and D. Greenberg. Perceptual illumination components: a new approach to efficient, high quality global illumination rendering. 23(3):742–749, 2004.
- [122] V. Sundstedt, K. Debattista, P. Longhurst, A. Chalmers, and T. Troscianko. Visual attention for efficient high-fidelity graphics. In *Proceedings Spring Conference on Computer Graphics*, pages 162–168, 2005.
- [123] V. Sundstedt, D. Gutierrez, O. Anson, F. Banterle, and A. Chalmers. Perceptual rendering of participating media. *ACM Transactions on Applied Perception*, 4, November 2007.
- [124] R. Tamstorf and H. W. Jensen. Adaptive smpling and bias estimation in path tracing. In *Proceedings of the Eurographics Workshop on Rendering Techniques*, pages 285–296, 1997.
- [125] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358, 1995.
- [126] W. B. Thompson, R. W. Fleming, S. H. Creem-Regehr, and J. K. Stefanucci. *Visual perception from a computer graphics perspective*. CRC Press, 2011.
- [127] D. Tian and G. AlRegib. Fqm: a fast quality measure for efficient transmission

- of textured 3d models. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 684–691. ACM, 2004.
- [128] F. Torkhani, K. Wang, and J.-M. Chassery. A curvature-tensor-based perceptual quality metric for 3d triangular meshes. *Machine GRAPHICS and VISION*, *In Press*.
- [129] F. Torkhani, K. Wang, and J.-M. Chassery. A curvature tensor distance for mesh visual quality assessment. In *Proceedings Computer Vision and Graphics*, pages 253–263. Springer, 2012.
- [130] C. J. Van den Branden Lambrecht and O. Verscheure. Perceptual quality measure using a spatiotemporal model of the human visual system. In *Electronic Imaging: Science & Technology*, pages 450–461, 1996.
- [131] L. Váša and J. Rus. Dihedral angle mesh error: a fast perception correlated distortion measure for fixed connectivity triangle meshes. *Computer Graphics Forum*, 31(5):1715–1724, 2012.
- [132] E. Veach and L. J. Guibas. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 65–76. ACM Press/Addison-Wesley Publishing Co., 1997.
- [133] C. T. Vu, E. C. Larson, and D. M. Chandler. Visual fixation patterns when judging image quality: Effects of distortion type, amount, and subject experience. In *IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 73–76, 2008.

- [134] B. Walter, S. Fernandez, A. Arbree, K. Bala, M. Donikian, and D. P. Greenberg. Lightcuts: a scalable approach to illumination. 24(3):1098–1107, 2005.
- [135] B. Walter, S. N. Pattanaik, and D. P. Greenberg. Using perceptual texture masking for efficient image synthesis. *Computer Graphics Forum*, 21(3):393–399, 2002.
- [136] K. Wang, F. Torkhani, and A. Montanvert. Technical section: A fast roughness-based approach to the assessment of 3d mesh visual quality. *Computer Graphics*, 36(7):808–818, Nov. 2012.
- [137] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [138] A. B. Watson, R. Borthwick, and M. Taylor. Image quality and entropy masking. *Human Vision, Visual Processing, and Digital Display VIII*, 3016, 1997.
- [139] B. Watson, A. Friedman, and A. McGaffey. Measuring and predicting visual fidelity. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 213–220. ACM, 2001.
- [140] T. Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, June 1980.
- [141] J. Wolfe, S. Butcher, C. Lee, and M. Hyle. Changing your mind: On the contributions of top-down and bottom-up guidance in visual search for feature

- singletons. *Journal of Experimental Psychology: Human Perception and Performance*, 29(2):483–452, 2003.
- [142] J. Wu, W. Lin, G. Shi, and A. Liu. Perceptual quality metric with internal generative mechanism. *IEEE Transactions on Image Processing*, 22(1):43–54, 2013.
- [143] M. C. Y. Xu. Selecting and perceiving multiple visual objects. *Trends in Cognitive Science*, 13(4):167–174, 2007.
- [144] H. Yee, S. Pattanaik, and D. P. Greenberg. Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. *ACM Transactions on Graphics*, 20(1):39–65, 2001.
- [145] M. Zhang, W. Wang, H. Sun, and H. Han. Perception-based model simplification for motion blur rendering. *Graphical Models*, 76(3):116–127, 2014.
- [146] Y. Zhao, L. Yu, Z. Chen, and C. Zhu. Video quality assessment based on measuring perceptual noise from spatial and temporal perspectives. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(12):1890–1902, 2011.
- [147] Q. Zhu, J. Zhao, Z. Du, and Y. Zhang. Quantitative analysis of discrete 3d geometrical detail levels based on perceptual metric. *Computers & Graphics*, 34(1):55–65, 2010.
- [148] A. Zotos, K. Mania, and N. Mourkoussis. A schema-based selective rendering framework. In *Proceedings of the 6th Symposium on Applied Perception in Graphics and Visualization*, pages 85–92, 2009.

List of Published Work

Journal Papers

1. L. Dong, W. Lin, Y. Fang, S. Wu, and H. S. Seah. Saliency detection in computer rendered images based on object-level contrast, *Journal of Visual Communication and Image Representation*, 25(3):525-533, 2014 (A part of Chapter 4 in this thesis is based upon this paper)
2. L. Dong, Y. Fang, W. Lin, C. Deng, C. Zhu, and H. S. Seah. Graphic image rendering based on an irregularity measure, *Signal Processing: Image Communication*, 33: 1-13, 2015 (A part of Chapter 5 in this thesis is based upon this paper)
3. L. Dong, Y. Fang, W. Lin, and H. S. Seah. Perceptual quality assessment for 3D triangular mesh based on curvature, *IEEE Transaction on Multimedia*, 17(12):2174-2184, 2015 (A part of Chapter 6 in this thesis is based upon this paper)

Conference Papers

1. L. Dong, W. Lin , C. Zhu and H. S. Seah, Joint Visual Attention and Rendering Complexity based Sample Rate Estimation in Selective Rendering, in *Proceedings IEEE International Symposium on Consumer Electronics*, pp.183-187, 2011 (A part of Chapter 3 in this thesis is based upon this paper)
2. L. Dong, W. Lin, C. Deng, C. Zhu, and H. S. Seah, Selective Rendering with Graphical Saliency Model, in *Proceedings IEEE Image, Video, and Multidimensional Signal Processing Technical Committee (IVMSP) workshop on Perception and Visual Signal Analysis*, pp.159-164, 2011 (A part of Chapter 3 in this thesis is based upon this paper)
3. L. Dong, W. Lin, C. Deng, C. Zhu, and H. S. Seah, To exploit uncertainty masking for adaptive image rendering, in *Proceedings IEEE International Symposium on Circuits and Systems (ISCAS)*, pp.2848-2851, 2013 (A part of Chapter 5 in this thesis is based upon this paper)
4. L. Dong, W. Lin, Y. Fang, S. Wu, and H. S. Seah, Detection of salient objects in computer synthesized images based on object-level contrast, in *Proceedings International Conference on Visual Communications and Image Processing (VCIP)*, pp.1-6, 2013 (A part of Chapter 4 in this thesis is based upon this paper)

5. L. Dong, Y. Fang, W. Lin, and H. S. Seah. Objective visual quality assessment for 3D meshes, *International Workshop on Quality of Multimedia Experience (QoMEX)*, pp. 1-6, 2014 (A part of Chapter 6 in this thesis is based upon this paper)