VISUAL ATTENTION MODELS AND APPLICATIONS TO 3D COMPUTER GRAPHICS

A DISSERTATION SUBMITTED TO
THE DEPARTMENT OF COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BİLKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Muhammed Abdullah Bülbül
June, 2012

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Assist. Prof. Dr. Tolga Çapın (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Prof. Dr. Bülent Özgüç

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Assist. Prof. Dr. Hüseyin Boyacı

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Assoc. Prof. Dr. Uğur Güdükbay

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Assist. Prof. Dr. Ahmet Oğuz Akyüz

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

VISUAL ATTENTION MODELS AND APPLICATIONS TO 3D COMPUTER GRAPHICS

Muhammed Abdullah Bülbül
Ph.D in Computer Engineering
Supervisor: Assist. Prof. Dr. Tolga Çapın
June, 2012

3D computer graphics, with the increasing technological and computational opportunities, have advanced to very high levels that it is possible to generate very realistic computer-generated scenes in real-time for games and other interactive environments. However, we cannot claim that computer graphics research has reached to its limits. Rendering photo-realistic scenes still cannot be achieved in real-time; and improving visual quality and decreasing computational costs are still research areas of great interest.

Recent efforts in computer graphics have been directed towards exploiting principles of human visual perception to increase visual quality of rendering. This is natural since in computer graphics, the main source of evaluation is the judgment of people, which is based on their perception. In this thesis, our aim is to extend the use of perceptual principles in computer graphics. Our contribution is two-fold: First, we present several models to determine the visually important, salient, regions in a 3D scene. Secondly, we contribute to use of definition of saliency metrics in computer graphics.

Human visual attention is composed of two components, the first component is the stimuli-oriented, bottom-up, visual attention; and the second component is task-oriented, top-down visual attention. The main difference between these components is the role of the user. In the top-down component, viewer's intention and task affect perception of the visual scene as opposed to the bottom-up component. We mostly investigate the bottom-up component where saliency resides.

We define saliency computation metrics for two types of graphical contents. Our first metric is applicable to 3D mesh models that are possibly animating, and it extracts saliency values for each vertex of the mesh models. The second metric

we propose is applicable to animating objects and finds visually important objects due to their motion behaviours. In a third model, we present how to adapt the second metric for the animated 3D meshes.

Along with the metrics of saliency, we also present possible application areas and a perceptual method to accelerate stereoscopic rendering, which is based on binocular vision principles and makes use of saliency information in a stereoscopic rendering scene.

Each of the proposed models are evaluated with formal experiments. The proposed saliency metrics are evaluated via eye-tracker based experiments and the computationally salient regions are found to attract more attention in practice too. For the stereoscopic optimization part, we have performed a detailed experiment and verified our model of optimization.

In conclusion, this thesis extends the use of human visual system principles in 3D computer graphics, especially in terms of saliency.

Keywords: Computer Graphics, Visual Perception, Saliency, Visual Attention, Binocular Vision, Stereoscopy, Motion Perception.

ÖZET

3-B BİLGİSAYAR GRAFİKLERİNDE GÖRSEL DİKKAT MODELLEMELERİ VE UYGULAMALARI

Muhammed Abdullah Bülbül
Bilgisayar Mühendisliği, Doktora
Tez Yöneticisi: Assist. Prof. Dr. Tolga Çapın
Haziran, 2012

3-B bilgisayar grafikleri, gelişen teknolojik imkanların da etkisiyle, çok yüksek seviyelere ulaştı ve günümüzde gerçeğe oldukça yakın görüntüler, bilgisayar oyunları ve diğer kullanıcı etkileşimi içeren uygulamar, gerçek zamanlı olarak üretilebiliyor. Fakat bilgisayar grafikleri alanındaki araştırmaların limitlerine ulaştığını iddia edemeyiz. Foto gerçekçi görüntüleme halen gerçek zamanlı olarak başarılamamakta olup, görsel kaliteyi artırmak ve görüntüleme maliyetlerini azaltmak araştırma alanı olarak ilgi odağı olmayı sürdürmektedir.

Son zamanlarda bilgisayar grafikleri alanındaki uğraşlar, görüntüleme kalitesini artırmak amacıyla görsel algı prensiplerini kullanmaya yöneldi. Bu, bilgisayar grafiklerinde, temel değerlendirme kriterinin insanların yargıları ve dolayısıyla algıları olmasının doğal bir sonucu. Bu tezde hedefimiz, görsel algı prensiplerinin bilgisayar grafikleri için kullanımını artırmaktır. Bu tezin literatüre katkısı iki alanda incelenebilir: Birincisi, 3-B sahnelerde görsel olarak önemli, dikkat-çeker, kısımları tespit etmeye yönelik sunulan modeller; ikincisi de, dikkat-çekerlik ölçütlerinin bilgisayar grafiklerinde kullanımına yapılan katkılar.

İnsanlarda görsel dikkat mekanizmasının iki kısmı vardır. İlk kısım, uyaranlara bağlı, aşağıdan yukarıya görsel dikkat olup; ikinci kısım göreve bağlı, yukarıdan aşağıya görsel dikkat olarak adlandırılmaktadır. Bu iki kısım arasındaki en önemli fark izleyicinin rolüdür. Yukarıdan aşağıya görsel dikkatte, aşağıdan yukarıya dikkatten farklı olarak, izleyicinin niyeti ve görevi sahnenin nasıl algılandığını etkiler. Çalışmalarımızda daha çok, içerisinde dikkat-çekerliği de barındıran aşağıdan yukarıya görsel dikkati araştırdık.

İki türlü grafiksel içerik türü için dikkat-çekerlik ölçütleri tanımladık. İlk ölçüt, 3-B hareketli grafiksel modeller için geliştirilmiş olup, modelin her

düğümüne bir dikkat-çekerlik değeri atamaktadır. İkinci model ise birden çok nesne barındıran bir animasyon sahnesinde, hareketlerinden dolayı görsel olarak önemli hale gelen nesneleri tespit etmeye yöneliktir. Üçüncü bir model de ise, ikinci modelde önerilen modelin ilk modelde kullanılan grafiksel içeriklere nasıl uygulanacağı gösterilmiştir.

Tezde, dikkat-çekerlik ölçütlerinin yanısıra, ölçütlerin muhtemel kullanım alanları ve ikili (stereo) görüntüleme için algıya bağlı bir optimizasyon yöntemi de sunulmuştur. Bu yöntem ikili görme prensiplerine dayanmakta olup, sahnenin dikkat-çekerlik bilgisinden yararlanmaktadır.

Sunulan yöntemlerin her biri, deneyler vasıtasıyla değerlendirildi. Dikkat-çekerlik ölçütlerinin değerlendirmesinde göz takip cihazı kullanıldı ve dikkat-çeker olarak belirtilen kısımlara daha çok bakıldığı tespit edildi. İkili görüntüleme için önerilen yöntem de, ayrıntılı bir kullanıcı testi ile doğrulandı.

Sonuç olarak, sunulan tez görsel sisteme dair prensiplerin, özellikle dikkat-çekerlik ile ilgili olanların, 3-B bilgisayar grafiklerinde kullanımını genişletmektedir.

Anahtar sözcükler: Bilgisayar Grafikleri, Görsel algı, Dikkat-çekerlik, Görsel dikkat, İkili Görüntüleme, Hareket algısı.

Acknowledgement

I know that the achievements we reached are granted gifts. As for the other parts of my life, I am really thankful to be greatly supported through the way of obtaining a Ph.D degree.

I would like to express my deepest gratitute to my supervisor Tolga Çapın for his kindness, great guidance, and support during my Ph.D study.

I would also like to thank my jury members, Bülent Özgüç, Hüseyin Boyacı, Uğur Güdükbay, and Ahmet Oğuz Akyüz; for their valuable comments and suggestions. They have sincerely helped and encouraged me through this study.

Thanks to all my friends, I really could not have achieved this much without their help and support. They formed a superb living environment in Bilkent.

Lastly, I thank each member of my family, especially to my wife Gamze. I couldn't think of a better support and love than they have given to me.

I would also like to acknowledge 3DPhone project and TUBITAK for financially supporting me during my Ph.D study.

Thanks a lot everyone.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In the last decade, the rendering and modeling methods in computer graphics advanced to very high levels, and it is now possible to generate very realistic synthetic scenes and animations including naturally behaving and looking simulations of fluids, humans, trees etc. Therefore, recent efforts of computer graphics researchers have directed towards accomplishing the generation of these high quality content in real-time for interactive applications or employing new methods that increase scene understanding, in addition to searching for more realistic modeling and rendering techniques.

In a very realistic computer generated scene, which may take hours to render despite the advances in technology, we do not recognize many details for which a notable amount of rendering effort is spent. One of the reasons behind this is our visual system's capacity to perceive detail. Additionally, the human eye only see two degrees of visual field, a little more than width of thumb at arms length, in high detail [124]. For the peripheral region, the resolution that we can perceive is much lower. However, we perceive all of the visual field in high quality, i.e. we do not feel any rendering artifacts in the generated image. This is achieved by rapid eye movements to visually important regions on our visual field and

combining the gathered information in our brain. Our visual system does not spend effort for insignificant details in a scene. Thanks to this behavior of human visual system, we could see the world in real-time. A similar approach could work well for computer graphics too. The principles of human visual attention mechanism could be exploited for various purposes in computer graphics field such as rendering optimization, modeling, and quality estimation.

Three dimensional (3D) stereo perception is also an area of potential interest in computer graphics research. Providing stereo imagery via stereo displays or glasses is quite an old idea. For example, Wheatstone invented a stereoscope to show slightly different images for each eye to provide binocular vision in 1838 [127]. Having been established long time ago, 3D imaging hasn't been used in a widespread fashion until recently. In recent years, however, the technology, towards 3D displays and rendering techniques and usage of binocular systems in movies and computer generated visuals, enhanced significantly. Usage of stereo vision brings along many challenges in addition to significantly increasing the generation times of visuals. Generating natural looking and comfortable 3D scenes is not an easy task [86].

To overcome the challenges that emerge in 3D computer graphics rendering, we need a good understanding and use of perception. Perception is of great importance since whatever happens in the world, our awareness of the events depends on our perception of them. Similarly, in computer graphics, success of any content rendered on the screen depends on our perception. Therefore, this thesis aims to make use of perceptual principles for generating computer graphics scenes that are perceptually in good quality and computationally less expensive.

Visual perception is a very well studied area in cognitive sciences and visual principles are studied for many centuries. For example, how the binocular vision mechanism works was an area of research in the 16th century [56]. Despite the advances in both visual perception and computer graphics, there is a need for searching new ways of incorporating them.

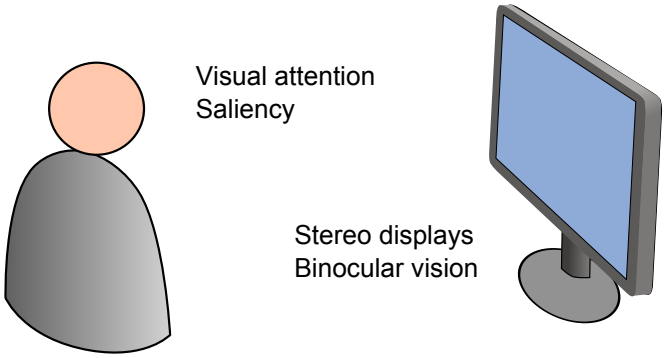


Figure 1.1: Aspects of our study

1.2 Scope of the Work

In this thesis, our main aim is to perform research on how to employ the perceptual principles in computer graphics and enhance computer graphics techniques on generating a perceptually aware system. Visual perception is a very large area including space perception, depth and size perception, motion perception, color perception and many more. Under each of these areas, there are numerous theories and studies in psychology, neuroscience, and computer science disciplines. Therefore, there is a need to restrict the scope of the efforts to concrete aspects.

In this respect, our study on the use of perceptual principles is motivated by two aspects which are illustrated in Figure 1.1. In both aspects, human visual attention and cognition mechanism constitute the basis and the common term in all aspects is the use of saliency, which characterizes the level of significance of objects/regions in our visual field. Therefore, this thesis focuses on saliency and its applications to computer graphics.

Identifying regions in a 3D scene that are visually more important for the user is the main concern in our study related to visual attention models. Therefore, this part utilizes the human visual attention mechanism and proposes several metrics to find out the visually attractive (salient) regions of various 3D graphical contents.

The attentional mechanism in humans has two components: Top-down (task

oriented) and bottom-up (stimulus driven). It is known that task and prior experiences bias the attentive process to the visual stimuli significantly. This type of attention constitutes the top-down component of attention which mostly depends on the viewer. On the other hand, the visual properties of a scene are also important on attracting the viewer and determining his gaze point. This second type of attention is purely stimulus driven and constitutes the bottom-up component of attention. Depending on the visual and temporal properties of objects in a scene, saliency resides in the bottom-up part. Since it is an impossible task to categorize all possible tasks and prior experiences of the user, our main concern here is the bottom-up part, which is mostly related to finding out the salient regions of a scene.

Another aspect of our study aims better use of displays providing faster 3D stereoscopic rendering without sacrificing visual quality. For stereoscopic vision systems that provide different images to each eye, we utilize the binocular vision principles of the human visual system. In a stereoscopic vision system, we analyze the response of the human visual system to the case in which right and left eyes are shown images, generated by different rendering parameters. We also investigate the relationship of such a rendering approach with visual attention mechanism and saliency.

1.3 Contributions

The contributions of this thesis are divided into two main parts: contributions on saliency computation and contributions on stereoscopic rendering optimization.

- A general saliency computation framework is proposed for animated meshes. The saliency framework makes use of the geometry, material, and motion properties of the meshes to extract their perceptually important regions. Per-vertex saliency calculation, which is performed in 3D space, enables view-point independent usage of the calculated saliency values. Possible application areas that use saliency values are also presented in the thesis.

- The second saliency based method extracts the saliencies of separate objects due to their motion. While the previous framework finds per-vertex saliency values, this method calculates saliencies on a per-object basis. Both of the studies are verified by formal experiments using an eye-tracker.

- Another contribution of the thesis is related to perceptual optimization of stereoscopic rendering. A mixed quality rendering method for views belonging to left and right eyes is presented. The suitability of important graphical methods are analyzed for this type of optimization and a general inference is obtained. For this study, a detailed experimental study is performed and the proposed technique is verified. The proposed technique helps to decrease stereoscopic rendering time notably.

  While these aspects form the main base of our study, there are additional benefits which will also be described in this thesis. Perceptual concepts and principles that are utilized in our studies are presented in detail. This literature is presented in three categories: visual attention mechanism, motion perception, and binocular vision.

1.4 Thesis Organization

The thesis is organized as follows: First, the utilized concepts and background information are presented and then the technical contributions and the proposed methods are given. After explaining the proposed methods, experimental evaluation of the studies are presented. The more precise outline of the thesis is as follows.

In Chapter 2, background information and the related studies are given in three categories: Visual attention, 3D vision, and motion perception. Additionally, current quality assessment means for 3D graphical models are given in this chapter.

Chapter 3 presents the proposed methods related to visual attention. These studies aim to extract the parts of the rendered scenes that capture the user's

attention. Saliency calculation metrics are presented in this chapter.

Stereoscopic vision aspect of our study, which aims saliency-guided perceptual optimization of stereoscopic rendering, is presented in Chapter 4.

In Chapter 5, the experiments to evaluate the proposed techniques and discussion of the results are demonstrated. Each part of our study is experimentally analyzed in separate sections.

Finally, we conclude the thesis and point out the possible future research directions in Chapter 6.

Chapter 2

Background

In this chapter, fundamental concepts utilized in the thesis and a review of the related literature on perceptually oriented computer-graphics research are presented in four sections. Firstly, visual attention mechanism in humans is presented which forms the main concepts used in Chapter 3 e.g., saliency and spatiotemporal sensitivity. Binocular vision mechanism and a review of stereoscopic rendering systems is given in Section 2.2 which are mostly related to the employed principles in Chapter 4. In Section 2.3, motion perception is presented. How to assess visual quality of the presented 3D content is an important part of our research. Lastly, a review of current literature on this area is presented.

This chapter is presented by a computer graphics perspective. The perception literature survey dealing with the details on how brain works, which parts of the visual cortex are employed in the visual perception etc. exceeds our scope of interest.

2.1 Visual Attention, Saliency, and Sensitivity

Seeing is an interaction between the objects we see and our vision system including our eyes and brain. Although there are many objects inside our periphery of

vision, some regions attract our attention more than others. The properties of the objects that we see, e.g., their sizes, motions, colors etc.; our intention of viewing, e.g., what are we looking for; and our prior experiences play a very important role for determining our gaze point.

While the direction of our attention in a visual scene is of great interest, how much detail we can perceive depends on another factor, visual sensitivity. For example, a very rapid movement may get attention but we are not sensitive to the details of this too quickly moving region, which could be utilized for optimization purposes in the field of computer graphics. Therefore, this section contains concepts related to our visual attention mechanism and visual sensitivity along with their realizations in the computer graphics field.

2.1.1 Concepts in Visual Attention

2.1.1.1 Bottom-up vs. Top-down

Visual attention mechanism could be divided into two components: bottom-up and top-down. Figure 2.1 illustrates these components.

Bottom-up part is related to object properties and is generally referred as stimulus-driven component of visual attention. As a simple example, in Figure 2.1, sizes of the boxes have an impact on attracting attention and the smaller one stands out among others as it is different. In the bottom-up part of the visual perception mechanism, intentional factors such as the user's task do not have an effect. It is mainly related to the visual and temporal properties of the objects. On the other hand, in the top-down part, task and prior experiences are main factors of the perception [61].

The interaction between the bottom-up and top-down components of attention could be explained as follows. The brain is firstly stimulated by objects in a bottom-up fashion, in which saliency has a great significance. Then, top-down intentional attention filters the scene according to the task of the observation and

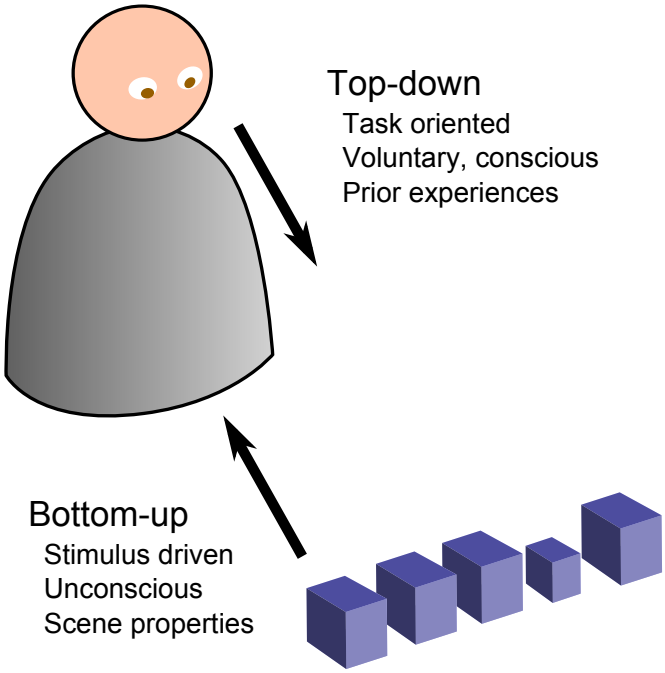


Figure 2.1: Two components of visual attention

experiences [102]. Both factors affect our perception of the visual scene and the direction of our gazes. The following sections summarize the details of bottom-up and top-down components of attention.

2.1.1.2 Bottom-up Component of Attention and Saliency

Certain properties of objects have an impact on driving our attention to specific regions in our visual periphery. The simplest example is an object on a plain background as shown in Figure 2.2. Compared to the background, the object attracts more attention and most probably becomes the first target of our gazes. We could say that the object is salient in this scene.

Visual saliency is a key concept which refers to the attractiveness of a visual stimulus for our visual system caused by its visual properties, e.g., size, shape, orientation, and color. It has been a focus of cognitive sciences for more than 20 years. Throughout the thesis, when we use saliency term we mean visual saliency unless otherwise stated.

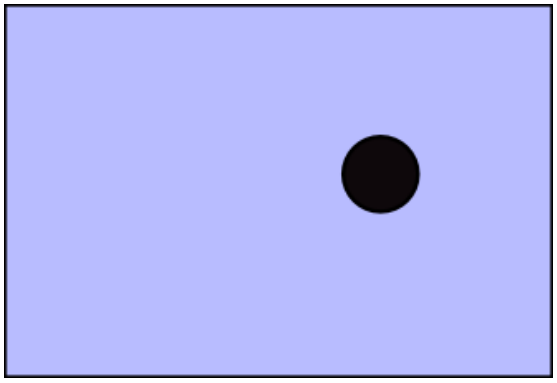


Figure 2.2: The object (black circle) is more salient than background and it attracts more attention.

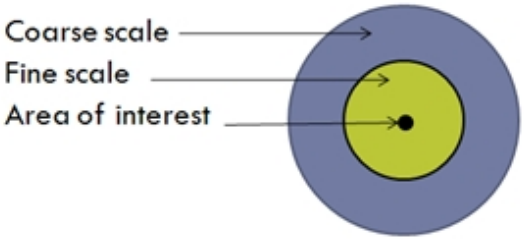


Figure 2.3: Center-surround mechanism, saliency of interested area is related to difference of fine and coarse scales in terms of different properties such as luminance, velocity, orientation, etc.

Bottom-up component of our visual attention is driven merely by the properties of the visual scene, disregarding user's intention while viewing the image. We could say that viewer independent factors (regardless of personal tasks, experiences etc.) determining the direction of our attention reside in the bottom-up part as saliency does.

Saliency is mainly related to difference of various visual properties of an object from its surroundings. The neurons employed in the visual system respond to image differences between a small central region and a larger surround region [61], which is known as the center-surround mechanism (Figure 2.3). This way, difference of a property compared to its surroundings stimulates us.

If an object is notably different compared to its surroundings it becomes salient. This difference could be in terms of many properties of the object, e.g.,

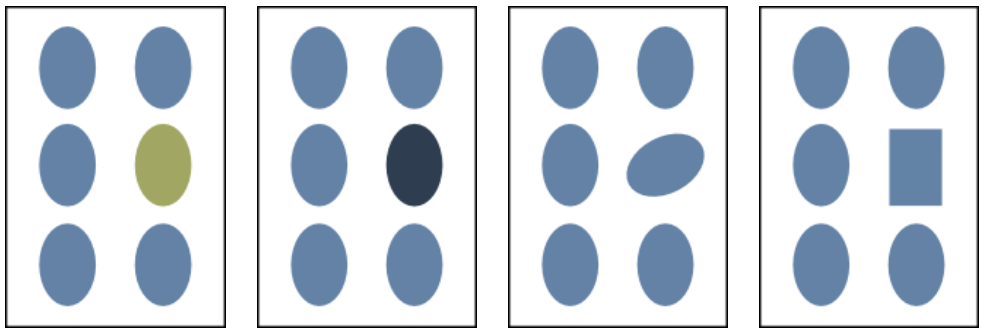


Figure 2.4: Difference in various properties affects saliency. Here, differing properties are hue (a), luminance (b), orientation (c), and shape (d).

hue, luminance, orientation, motion etc. (See Figure 2.4). It could be said that saliency is mostly related to difference of a property rather than the strength of it; e.g., we cannot say that a specific color makes a region always salient.

A highly salient object pops out from the image and immediately attracts attention. This process is unconscious and operated faster compared to the task oriented attention. The speed of bottom up (saliency based) attention is on the order of 25 to 50 ms per item, while the task oriented attention takes more than 200 ms [61]. Howlett et al. [58] show that faces of natural objects such as animals are salient compared to other parts. Besides, the existence of a special high-level mechanism for face perception in human visual system is a controversial issue. Hershler and Hochstein [49] [50] claim that there is a high level, possibly innate, mechanism in the visual system making faces pop out in an image. VanRullen [119] opposes to this claim in his study stating that there is a pop-out effect for faces but it is mostly based on low-level factors. Based on these studies we could say that faces do pop out but it is controversial if face perception resides in the bottom-up component or the top-down component of visual perception.

Hoffman and Singh [53], in their research for identifying the factors affecting the saliencies of the components of objects, conclude with the following findings. Firstly, 3D shapes are perceptually divided into separate parts on their concave creases. For the parts generated perceptually, larger size relative to the whole object and larger protrusion from the object cause higher saliencies (Figure 2.5);

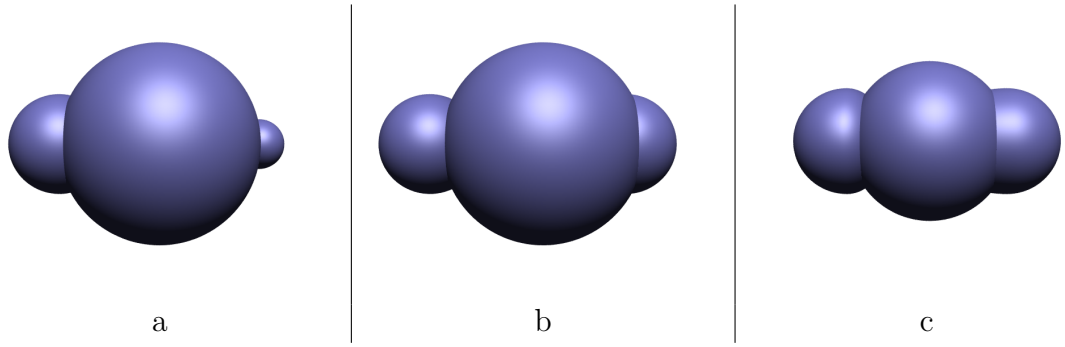


Figure 2.5: Saliency by parts: larger size, larger protrusion, and stronger boundaries (having higher crease angles) increase saliency of the part [53].

moreover, as well as the already visible relative size and visible protrusion of parts, their perceptually completed sizes and protrusions are also effective on their saliencies. Another finding is that a boundary with a higher curvature is more salient than a boundary with less curvature (Figure 2.5-c).

2.1.1.3 Top-down Component of Attention

What are we looking for greatly affects our visual perception. When we look for a specific type of object or for a specific property we could perceive many details that are not perceived normally. On the other hand, biasing the perception towards a specific target makes other objects less perceivable. Figure 2.6 presents the significant effect of task on determining our eye movements based on the results of the experiment of Yarbus [133].

This form of attention is called *top-down task oriented atttention* and is voluntary. Compared to the bottom-up involuntary attention, it is slower.

After being stimulated from the scene in a bottom-up fashion, goal oriented top-down attention determines what is perceived. This phase of attention includes constraining the recognized scene, based on scene understanding and object recognition [61]. When a scene is constrained by the visual system, the region which gets the most attention is promoted, which is known as the winner-take-all principle [61].

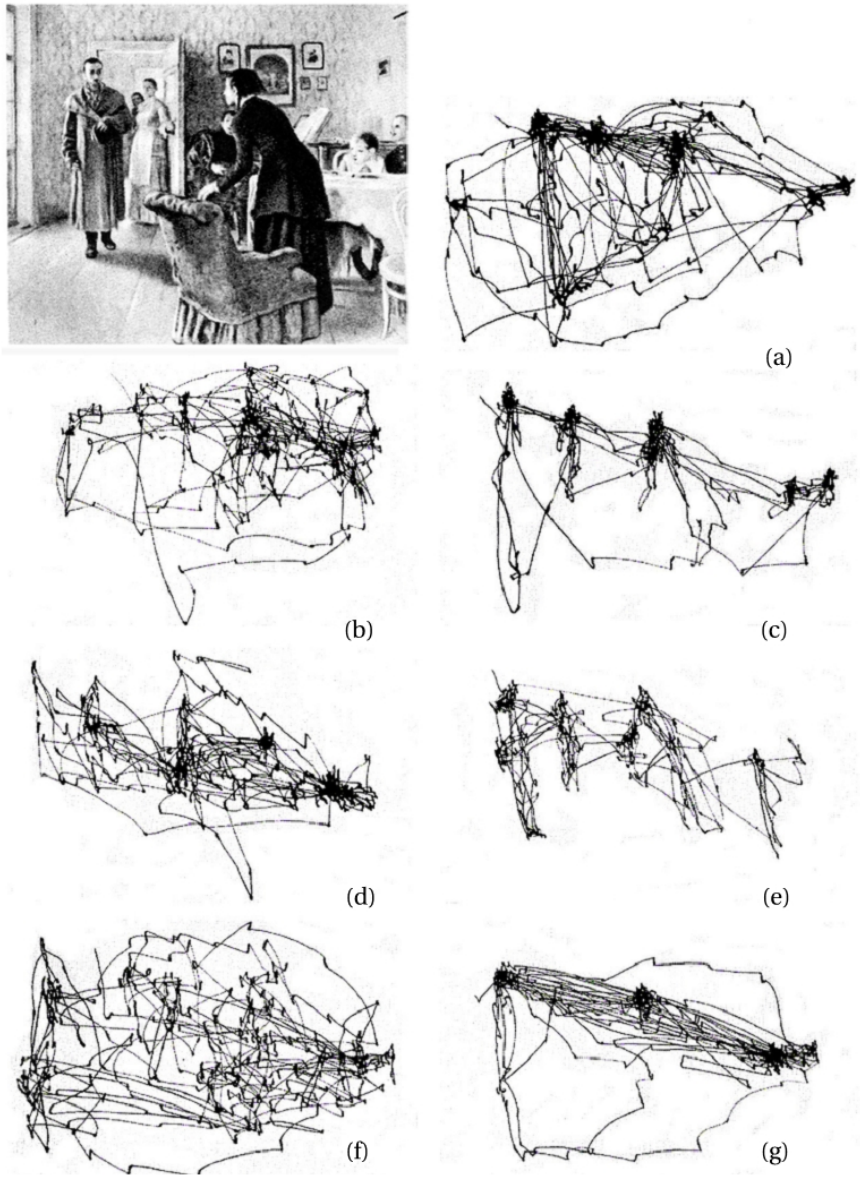


Figure 2.6: Repin's picture was examined by subjects with different instructions; (a) Free examination. (b) Estimate thematerial circumstances of the family in the picture. (c) Give the ages of the people. (d) Surmise what the family had been doing before the arrival of the 'unexpected visitor'. (e) Remember the clothes worn by the people. (f) Remember the position of the people and objects in the room. (g) Estimate how long the unexpected visitor had been away from the family. (From [115]. ©Benjamin W. Tatler, Nicholas J. Wade, Hoi Kwan, John M. Findlay, and Boris M. Velichkovsky; reprinted with permission.)

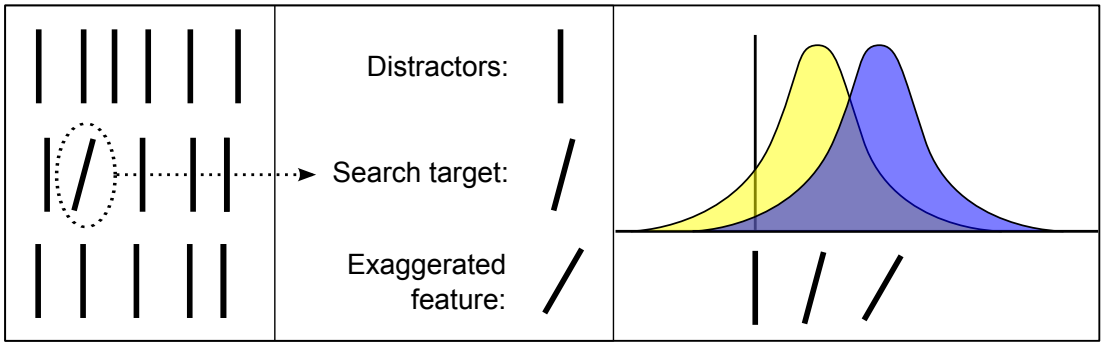


Figure 2.7: Human visual system is tuned to the exaggerated feature, which is a better discriminator, to optimize the search process.

With a search task on browsing a scene, the human visual system is tuned in the optimal way according to the search goal, such that the features of our target become easily recognizable [91]. Interestingly, our visual system is not adjusted for the exact features of our search target, but it is adjusted so that we could differentiate these features in the optimum way. For example, among objects that are oriented in a upwards direction, if our goal is to find a slightly right slanted object; our sensitivity is tuned for the exaggerated feauture of the target object to simplify differentiation (See Figure 2.7). In the same way, when our attention is tuned according to a search goal, we may not recognize the objects that are not related to our task although they are easily visible, which is called *inattentional blindness* [112].

Another principle of visual attention is inhibition of return, firstly described in 1984 by Posner and Cohen [99], which provides our visual system to perceive the entire scene instead of being stuck in the visually most attractive region. According to this principle, when a region is attended once, our perception on that region is inhibited after the first 0.3 seconds and the recognition of objects in this location decreases for a time of approximately 0.9 seconds. As a result, attention goes to a new region enabling search of different and novel regions in the visual periphery.

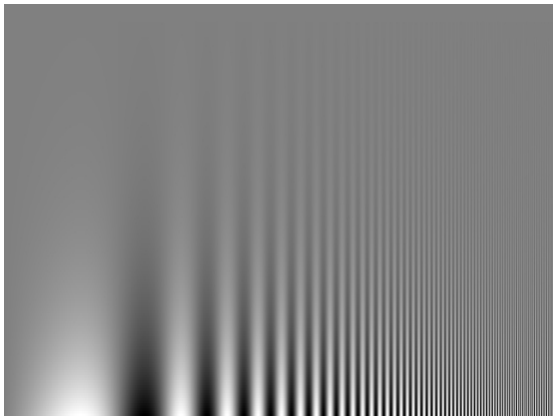


Figure 2.8: Campbell-Robson Contrast sensitivity function chart [22]. The frequency increases from left to right and contrast decreases from bottom to top. (Image from [94]. Courtesy of Izumi Ohzawa, reprinted with permission.)

2.1.1.4 Sensitivity

Our sensitivity to the details in a scene can be utilized in computer graphics. There are previous attempts that analyze the sensitivity of human visual system to the visual scene. Spatial and temporal frequencies of the scene significantly affect our sensitivity [65]. The general behaviour of sensitivity to spatial frequency could be seen in Figure 2.8. As shown in the figure, our sensitivity to contrast difference decreases in both ends of this figure. Additionally, there is an interaction between spatial and temporal frequencies. The way the sensitivity is affected by spatial and temporal frequencies is shown in Figure 2.9 [65].

The human visual system tolerates small velocities and could trace the objects as if they are static. The temporal frequencies shown in Figure 2.9 are according to the retinal velocities of the moving patterns. Daly [30] proposed a heuristic to compute the retinal velocity as:

$$v_R = v_I - min(0.82v_I + v_{min}, v_{max}), \tag{2.1}$$

where v_R is the retinal velocity, v_I is the velocity in image space, v_{min} is the velocity that the eye can track as if there is no motion, and v_{max} is the maximum velocity that the eye can effectively track. v_{min} and v_{max} are set to be $0.15°$/sec

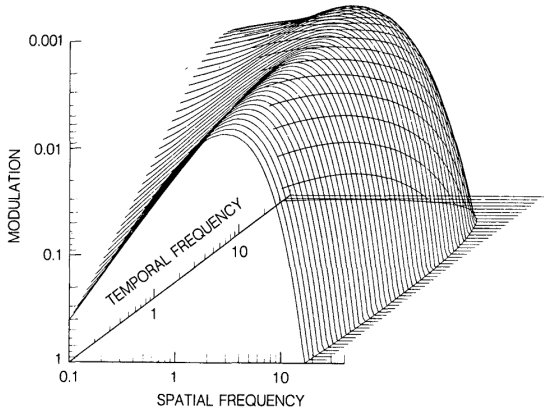


Figure 2.9: Spatiotemporal sensitivity formula derived by Kelly. (From [65]. ©1979 Optical Society of America, reprinted with permission.)

and 0.8°/sec by Daly.

While these functions are approximate sensitivity thresholds, when a signal exceeds the sensitivity threshold, it is not guaranteed to be perceived at each trial. In Psychophysics, the minimum luminance value that we can see is called the absolute threshold and the minimum luminance difference that we can perceive is called just noticable difference (JND). Absolute threshold can be measured as the minimum strength of a signal that is just discriminable from its null [37].

Our sensitivity to a signal could be decreased by the presence of another signal, which is called the *masking effect*. A simple example is the auditory masking effect: a sound could be less perceptible in the presence of a louder sound. Similarly, visual properties could have a masking effect. For example, texture on a 3D model could mask the artifacts on the model's surface. Figure 2.10 shows an example of masking effect for simplified 3D models. This type of masking is utilized in computer graphics to hide the low tessellation of the model by the use of textures [35]. Lavoué analyzed the masking effect of surface roughness on the perceived distortion of 3D models [70]. The distortions on the surface, e.g., noise and watermarking are found to be less perceptible on rough regions compared to smooth regions [70].

The eye can see sharply in only foveal region and it is less sensitive to detail in the peripheral region, although its sensitivity does not drop off to zero instantly

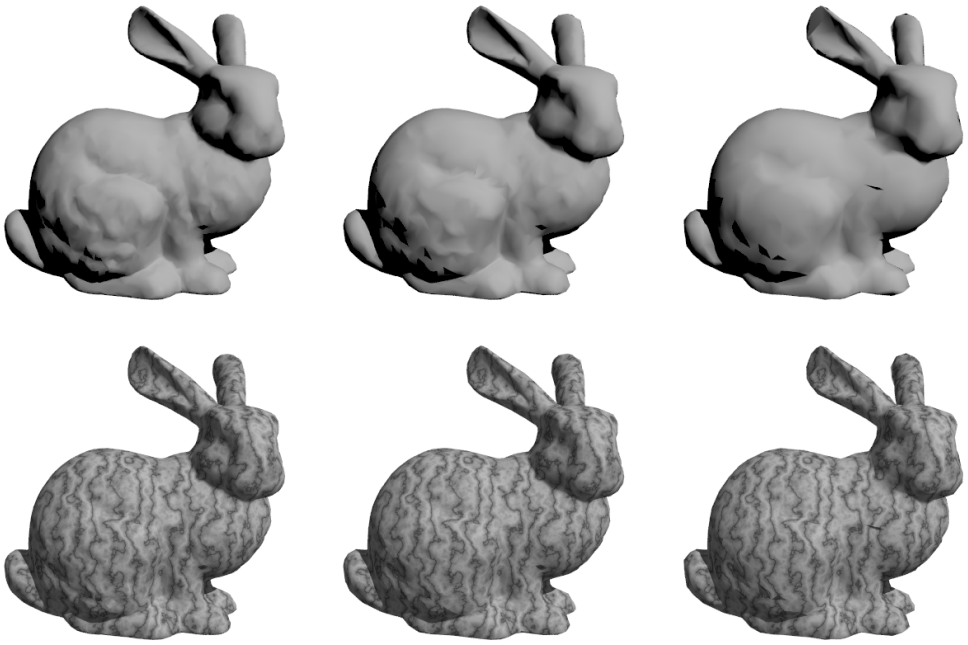


Figure 2.10: Masking effect due to textures: Simplification is less recognizable when texture is applied to the models. From left to right: the number of faces in the models are approximately: 8600, 4300, and 2150, respectively.

when going away from the center of interest [102]. While we could see the colors in sharp detail on the foveal image, the color perception decreases significantly for the periphery where we are more sensitive to luminance compared to color. The reason for this is the positions of the color sensitive cone cells and luminance sensitive rod cells on retina. Cones reside mostly in the fovea and more effective in the foveal region of our vision. Rods surround fovea and provide better luminance sensitivity in the peripheral region.

2.1.2 Visual Attention in Computer Graphics

2.1.2.1 Computational Models for Visual Attention and Saliency

Itti et al. [62] [61] describe one of the earliest methods to compute the saliency of two dimensional (2D) images (Figure 2.11-top). In order to calculate the saliency of a region, they compute the Gaussian weighted means of the intensity,

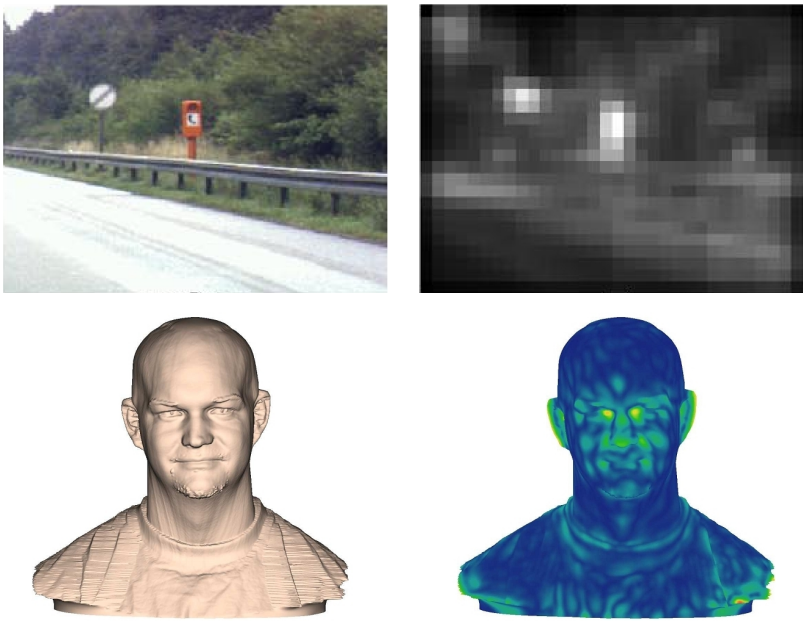


Figure 2.11: Saliency computation in 2D (top) by Itti et al. [62] (From [62]. ©1998 IEEE, reprinted with permission.) and in 3D (bottom) by Lee et al. [74] (From [66]. Courtesy of Youngmin Kim, reprinted with permission). In the saliency images bright regions represent more salient regions.

orientation, and color opponency properties in narrow and wide scales; then the difference of these scales gives the information of how different this region is compared to its surroundings.

Lee et al. [74] have introduced the concept of mesh saliency of 3D graphical models (Figure 2.11-bottom). In their work, the saliencies of mesh vertices are computed based on the mesh geometry. Their proposed mesh saliency metric is based on the center-surround operator on Gaussian weighted mean curvatures. They have used the computed saliency values to drive the simplification of 3D meshes, using Garland and Heckbert's [40] Qslim method for simplifying objects based on quadric error metrics.

Another saliency metric and measure for the degree of visibility is proposed by Feixas et al. [34]. Their saliency metric uses the Jensen-Shannon divergence of probability distributions by evaluating the average variation of JS-divergence between two polygons, yielding similar results as Lee et al. [74]. A saliency map for selective rendering that uses colors, intensity, motion, depth, edges, and

habituation (which refers to saliency reduction over time as the object stays on screen) is developed using GPU [80]. Their saliency map is based on the model suggested by Itti and Koch [60].

The mesh saliency metric was improved by Liu et al. [78]. In their work, two main disadvantages of Lee et al.'s work [74] are discussed. One is that the Gaussian-weighted difference of fine and coarse scales can result in the same saliency values for two opposite and symmetric vertices, because of the absolute difference in the equation. The other is that combining saliency maps at different scales makes it difficult to control the number of critical points. Therefore, instead of the Gaussian filter, they use a bilateral filter and define the saliency of a vertex as the Gaussian-weighted average of the scalar function difference between the neighboring vertices and the vertex itself.

2.1.2.2 Application Areas of Visual Attention and Saliency

Saliency and other perceptually-inspired metrics have gained attention in level-of-detail (LOD) rendering and mesh simplification. Reddy [102] uses the models of visual perception, including vision metrics such as visual spatial frequency and contrast, to optimize the visual quality of rendering for a flythrough in a scene, by removing the non-perceptible components of 3D scenes. Luebke and Hallen [83] propose a perceptually-driven rendering framework that evaluates local simplification operations according to the worst-case contrast gratings and the worst-case spatial frequency of features that they can induce in the image. In their work, contrast grating is a sinusoidal pattern that alternates between two extreme luminance values, and the worst-case one is a grating with the most perceptible combination of contrast and frequency induced by a simplification operation. They apply the simplification only if a grating with that contrast and frequency is not expected, so they do not get any perceptible effect, which results in a high fidelity model. A set of experiments have been performed using three groups of tasks for measuring visual fidelity [126]. These tasks are naming the model, rating the likeness of the simplified model against a standard one using a 7-point scale, and choosing the better model of two equally-simplified models

using Q-Slim and V-clust [106]. The results of these experiments and some automated fidelity measures [18] [25] show that automated tools are poor predictors of naming times but good predictors of ratings and preferences. Williams et al. [128] extend the perceptual simplification framework by Luebke and Hallen [83] to models with texture and light effects. Howlett et al. [57] use an eye tracker to identify salient regions and the fixation time on these regions of models, and they modify Q-Slim to simplify those regions with a weight value. Because of experiments similar to Watson et al.'s work, it is shown that the modified Q-Slim performs better on natural objects, but not on man-made artifacts, which indicates that saliency detection is very important.

Although mostly used for simplifying meshes, saliency has also been used as a viewpoint selection criterion. In Yamauchi et al.'s work [130] viewpoints are selected among a sample point set, forming the vertices of a graph on the bounding sphere of an object. The graph is partitioned according to the degree of similarity between its edges, and sorted according to their geometric saliency value. A recent work by Shilane et al. [110] uses the database of objects to measure the distinctiveness of different regions of an object. It is based on the idea that if a region has a very unique shape that is used to differentiate the object from other objects, that region is an important part of the object. It works by selecting several random points as centers of overlapping spheres over the surface and generating shape descriptors from the surfaces covered by those spheres. Next, a measurement is taken of how distinctive each region is with respect to a database of multiple object classes, and if the best matches of a region are all from the object's own class, that region is distinctive. Although a database is required, it gives better results than Lee et al's approach [74] in terms of simplification quality.

Saliency has also been studied for illustration. It is shown that visual attention can be directed by increasing the saliency at user-selected regions using geometric modification [67] (Figure 2.12). With a weight change in the center-surround mechanism, they modify mean curvature values of vertices by using bilateral displacements and use eye trackers to verify that the change increases user attention. Mortara et al. [89] use the saliency information to generate

Figure 2.12: Several applications utilizing saliency. Left: User's attention is directed to the second statue by altering its saliency [66] (Courtesy of Youngmin Kim, reprinted with permission). Right: Saliency information is used in generating cubist like paintings [10] (Courtesy of Sami Arpa, reprinted with permission).

thumbnails of meshes. In addition to the bottom-up saliency calculation they also use semantic information to determine the important parts of a mesh.

Saliency based variation of human models is proposed by McDonnell et al. [85]. In this work, various human models in a crowd are generated by modifying only the salient regions, namely head and upper torso, of the models. This eases crowd generation with perceptually different individuals.

Saliency information could also be used for artistic purposes (Figure 2.12). For example, saliency information is utilized to generate cubist like renderings [27] and also used in automatic caricature generation of 3D models [26].

Top-down component of attention is also used in computer graphics. For example, having the information of task related of objects could be utilized very efficiently in computer graphics by directing the rendering efforts on these objects, depending on the assumption that other objects will not get attention. Cater et al. [23] utilize this assumption to come up with a selective rendering framework. On the other hand, most of the time, it is not very practical to have the knowledge

of task related objects in a 3D environment. In another study, Lee et al. [75] study the attention given on objects that are tracked in real-time and uses it to adjust the level of detail. In this study, an interactive 3D environment which provides free user movement is used and the assumption is that the objects that are tracked by the user gets attention.

2.2 Binocular Vision and Stereoscopic Rendering

This section is divided into three parts: The first part gives the fundamentals of binocular vision and stereoscopic rendering, then the optimization methods for stereoscopic rendering in the literature are presented. Lastly, binocular suppression theory of binocular vision, which forms the base of our study on stereoscopic rendering optimization, is given.

2.2.1 Concepts in Binocular Vision

2.2.1.1 Binocular vision fundamentals

The human visual system extracts depth information via several cues of depth. Most of these cues, such as perspective, relative size, texture gradient, exist in monocular images 2.13. Although we could extract most of depth information from the 3D renderings in monocular displays today '3D display' is a phrase referring to displays capable of providing different images to right and left eyes, enabling binocular vision.

Stereo vision is a powerful depth cue, and when used properly it provides a strong presence feeling and 3D sense for small distance, i.e., effective for objects closer than 30m to the eye. When two eyes see slightly different images, the human vision system uses the disparities of objects in these two images to extract depth information and get a 3D impression, as shown in Figure 2.14.

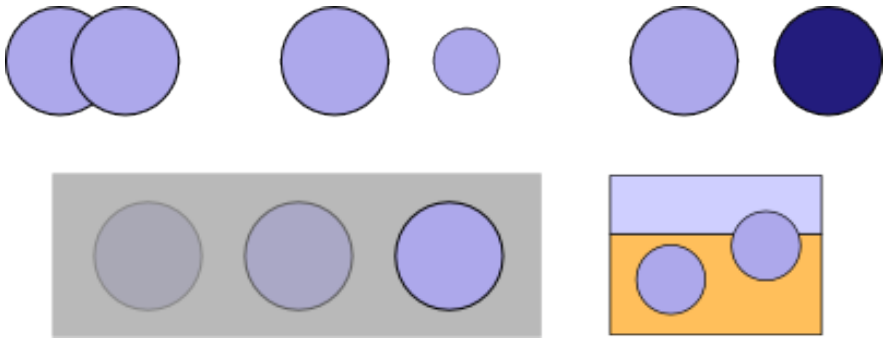


Figure 2.13: Several monocular depth cues including: occlusion, relative size, relative brightness, atmosphere, and distance to horizon.

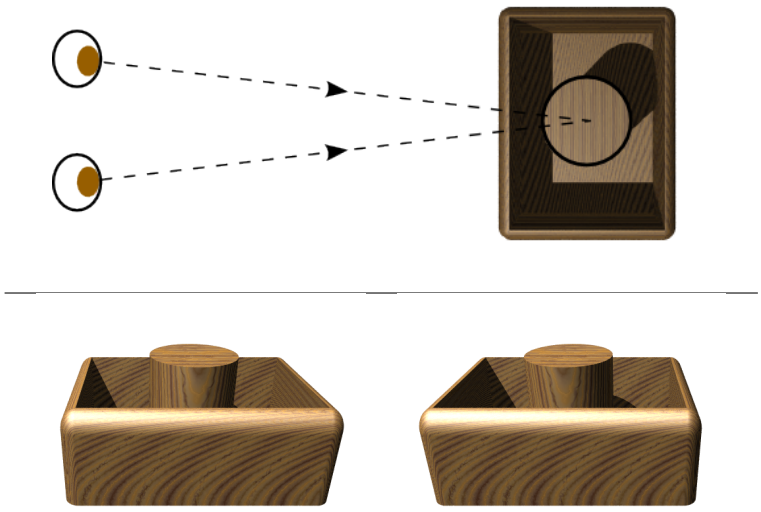


Figure 2.14: Visual system uses the difference of images viewed by left and right eyes to extract depth information.

Binocular vision increases visual workload [87], requiring analysis of two different views and their relations. Long durations of 3D vision with contents having high disparity causes discomfort and eye fatigue. Similarly, in computer graphics, stereoscopic rendering requires rendering the scene twice and decreases rendering performance. The studies to optimize stereoscopic rendering are given in Section 2.2.2.

Convergence-accommodation conflict: A problem that emerges in stereo rendering systems is convergence-accommodation conflict. A single eye physically adapts to the distance of the focused object by distorting eye lens, that is known as accommodation and is a weak depth cue. Also, two eyes converge to the focal depth as illustrated in Figure 2.14, which is known as convergence (or vergence). In physical world, these two depth cues support each other. However, with the use of 3D displays, a single eye accommodates to the display distance while convergence is made according to the virtual depth of the scene due to the disparity of the shown left and right images, which causes a conflict about focal distance resulting in fatigue and visual discomfort [52].

2.2.1.2 Binocular Suppression Theory

Binocular suppression theory of binocular vision proposes an explanation to the binocular vision mechanism of the eye. According to this theory, when dissimilar images are shown to each eye, one of the views suppresses the other at any one time, and the dominating view alternates over time. But when similar images (e.g., in a stereo pair) are shown to each eye, similar images falling on corresponding retinal regions form a unitary visual impression, while each region in the visual field contains input from a single eye at any one time [56].

Even though the actual process of the binocular vision is not fully identified, there are cases which support the binocular suppression theory in perception research. For instance, in an experimental study, subjects were asked to wear a lens for myopia for one eye, and hyperopia for the second eye, and were observed to see all distances in sharp focus, because the focused image suppresses the

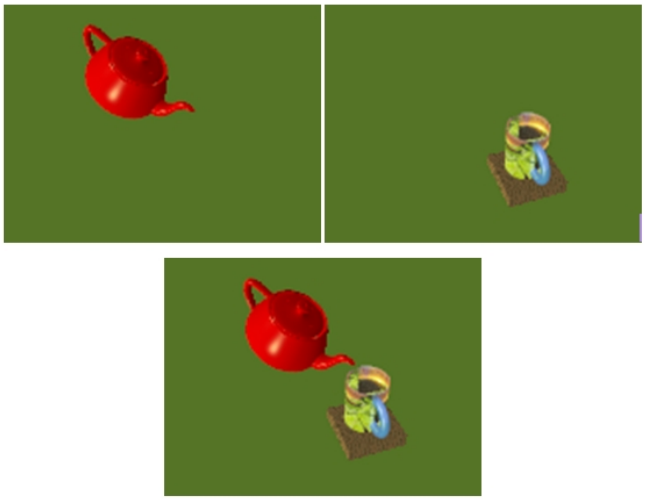


Figure 2.15: Binocular rivalry mechanism. When the left-eye (top-left) and right-eye (top-right) views are shown, the combined view (bottom) merges the dominant regions from the two views. (From [20]. ©2010 Elsevier, reprinted with permission.)

unfocused eye. This further supports the binocular suppression theory that one view is suppressed by the other, with no effect on the final percept [56].

According to the binocular suppression theory, when one view is suppressed by the other, a perceptual competition occurs between the two views. This is known as binocular rivalry and this property has been studied extensively. Asher [12] states that rivalry occurs in local regions of the visual field, and only one eye's view is dominant within these regions. Figure 2.15 illustrates this mechanism. In the combined view, the teapot and the glass completely suppress the corresponding portion of green ground seen by the other eye. Blake and Logothetis [17] also examined the principles of the binocular vision and claimed that stronger competitors have larger dominance. For instance, a high-contrast figure will dominate over a low-contrast one, or a brighter stimulus has an advantage over a dimmer one from the perspective of predominance.

Once binocular rivalry mechanism is confirmed, the next question becomes: what are the factors that affect the strength of a region for rivalry? Yang et al. state that a pattern with higher spatial frequency in one eye suppresses a pattern with lower spatial frequency in the other eye; therefore it is stronger [56].

Similarly, a region becomes stronger when the contrast [16] or the number of contours increase, which in turn cause a higher spatial frequency. Color variance also has a positive effect on stimulus strength [54]. One other factor that causes a stimulus to be stronger is motion [56]. According to Breese, a moving grating has an advantage over a stationary one and the strength increases as the speed of motion increases [56].

Binocular suppression theory has recently gained interest in the image processing and compression fields. Perkins [96] studied mixed-resolution stereo image compression where one view is low-pass filtered and has lower resolution, and demonstrated that the resultant 3D percept is of adequate image quality, when compared to the reference content. In a related work, Berthold [14] showed that apparent depth is relatively unaffected by spatially filtering both channels of a stereo image. Therefore, the image processing research to-date suggests that it is possible to low-pass filter one or both views of a stereo pair without affecting the subjective impression of sharpness, depth, and quality of the image sequence. Stelmach et al. [113] has built on these results, and presented a solution for mixed-resolution stereo image compression, and provided favorable experimental results.

2.2.2 Stereoscopic Rendering Optimization Techniques

A number of techniques have been proposed to optimize stereoscopic rendering. The first group of solutions follows a graphics pipeline-based approach, by utilizing the coherence between neighboring views. Adelson et al. [4] simultaneously render a triangle to both images by using the x-axis coherence in device coordinates to accelerate the stereoscopic rendering process. Kalaiah and Capin [63] propose a GPU-based solution that reduces the number of vertex shader computations needed for rendering multiple views, the vertex shader is split into two parts - view-independent and view-dependent. Performing vertex shader computations once for the view-independent part, instead of per-view calculation, reduces the rendering complexity. Hasselgren et al. [44] propose a multiview pipeline-based method, called approximate rendering, where fragment colors in all neighboring

views can be approximated from a central view when possible. As a result of approximate rendering, many per-pixel shader instructions are avoided.

Another group of solutions uses an image-based approach. In these solutions, one view is reconstructed from the other, 3D rendered view, by exploiting the similarity between the two views. In these techniques, the rendering time of the second image depends on only the image resolution, instead of the scene complexity, therefore saving rendering computations for one view. Fu et al. [39] compute the right image by warping the left image; however the resulting image contains holes which require to be filled by interpolation. Wan et al. [120] fill these holes by raycasting. Similarly, Fehn [33] uses a depth buffer to generate multiple views from a single image. Blurring the depth buffer by a Gaussian filter is used for handling the hole-filling problem. Zhang and Tam [135] also use depth images to generate the second view. In this method, the image for one view is used to construct a depth image, and then the second view is constructed using this depth image. Lastly the holes occurred in the previous step are filled by averaging the textures from neighboring pixels. Halle uses epipolar images that contain the rendered primitives interpolated between the two most extreme camera viewpoints for extracting the in-between views [43]. Stereo images produced with these techniques are generally an approximation to the original stereo image rendering result.

Finally, a third group of solutions has been proposed for stereoscopic rendering optimization targeted for ray tracing and volume rendering. Adelson and Hodges [4] propose a solution to stereoscopic ray tracing, where a ray-traced left image is used to construct part of the right image and the rest of the right image is calculated by ray tracing. He and Kaufman [45] speed up stereoscopic volume rendering by re-projecting the samples for the left view to the right image plane and compositing several samples simultaneously while raycasting. Similarly, Es and Isler [32] propose a GPU-based approach for efficient implementation of stereoscopic ray tracing.

2.3 Motion Perception

2.3.1 Concepts in Motion Perception

2.3.1.1 Mechanism to Perceive Motion

A difference of position in our visual field provides a sense of motion. This process requires a temporal analysis of the contents in our visual field. Between two different images that fell into our retina sequentially, our visual system needs to identify that some objects placed in different positions are the same objects and they are moving. Despite we could easily perceive objects as smoothly moving, the mechanism to detect motion is not that simple. Dealing with spatial relations is easier than solving temporal relations [38]. Also, it is possible not to be able to perceive motion, which is called as 'motion blindness'. Patients suffering from motion blindness could see moving objects as static ones that change their places abruptly [38].

A proposed model to explain motion detection is Reichardt motion detector [103]. This device is based on small units responsible for detecting motions in specified directions. These units compare two retinal image points, if the same signal appears in these two points with a small delay, the units detect motion in their specific direction [38]. Along with color, depth, and illumination; center-surround organization is also applied to motion processing in visual system. The neurons processing motion have a double-opponent organization for direction selectivity [8] meaning that the motion detecting modules could inhibit their surroundings and a motion should be differentiable compared to its surroundings to be detected.

2.3.1.2 Motion and Luminance

Motion perception has a close relation with the luminance channel. A strong contrast between the luminance values of the moving object and the background

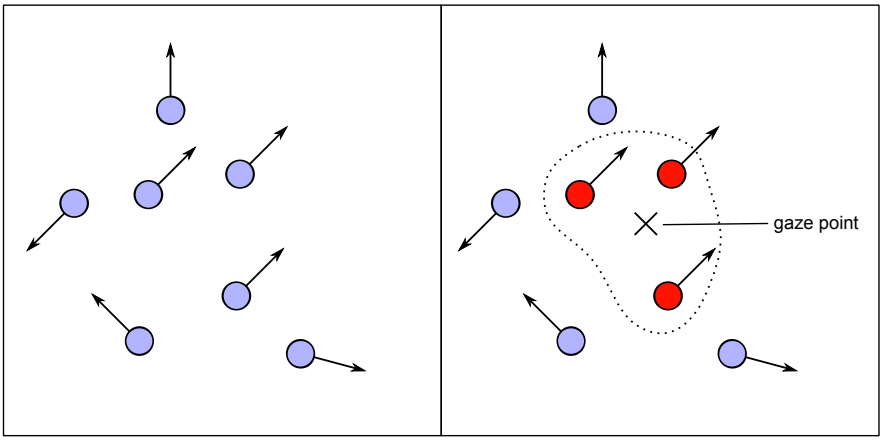


Figure 2.16: Left: arrows represent motion directions of the separate units. Right: According to Gestalt psychology, the units with the same motion behavior are united and perceived as a single unit. In such a case, viewers look at somewhere inbetween the group members instead of focusing on one of them.

enhances motion sensitivity. A pattern seems to be moving significantly slower when the background and foreground has only chromatic difference (with same luminance values) compared to the case of black and white [125]. Spatial frequency is also effective on temporal sensitivity as presented in Section 2.1.1.4.

2.3.1.3 Gestalt Psychology for Motion Perception

In spatial domain, visual system tends to group stimuli by considering their similarity and proximity as introduced in Gestalt principles. It is shown that visual system searches similarities also in temporal domain and can group stimuli by considering their parallel motions [68]. A bunch of moving dots with the same direction and speed could be perceived as a moving surface with this organization. Figure 2.16 illustrates this type of grouping.

2.3.1.4 Motion Aftereffect

Upon viewing a motion for a notable amount of time, when we look at somewhere else, a static scene appears to be moving in the opposite direction compared to the motion we have been seeing. This is called 'motion aftereffect' and it is the

result of our visual system's adaptation to a motion.

2.3.1.5 States of Motion

Visual motion may be referred as salient since it has temporal frequency. On the other hand, recent studies in cognitive science and neuroscience have shown that motion by itself does not attract attention. However, phases of motion, e.g., motion onset, motion offset, continuous motion, have different degrees of influence on attention. Hence, each phase of motion should be analyzed independently.

Abrams and Christ [3] experimented different states of motion to observe the most salient one. They indicated that the onset of motion captures attention significantly compared to other states. Immediately after motion onset, the response to stimulus slows with the effect of inhibition of return and attentional sensitivity to that stimulus is lost.

Singletons, having different motion than others within stimuli, capture attention in a bottom-up, stimulus-driven control. If there is a target of search, only feature singletons attract attention. If it is not the target, observers' attention is not taken. However, abrupt visual onsets capture attention even if they are not the target [132].

Other than motion onset, the experiments in the work of Hillstrom and Yantis [51] showed that the appearance of new objects captures attention significantly compared to other motion cues. On the other hand, motion offset and continuous motion doesn't capture much attention.

2.3.2 Motion Perception in Computer Graphics

Visual sensitivity to moving objects is utilized in computer graphics. Kelly [65] and Daly [30] measure the spatio-temporal sensitivity and fit computational models according to their observations. Yee et al. [134] built on these studies and used the spatio-temporal sensitivity to generate error tolerance maps to accelerate

rendering.

Peters and Itti [97] observed the gaze points on interactive video games and concluded that motion and flicker are the best predictors of the attended location while playing video games. Their heuristic for predicting motion-based saliency (as for other channels like color-based and orientation-based) works on 2D images and it is also based on the center-surround machanism.

Halit and Capin [42] proposed a metric to calculate the motion saliency for motion-capture sequences. In this work, the motion capture data is treated as a motion curve and the most salient parts of these curves are extracted as the keyframes of the animation.

2.4 Quality Assessment of 3D Graphical Models

This section presents recent advances in evaluating and measuring the perceived visual quality of 3D polygonal models. The general process of objective quality assessment metrics and subjective user evaluation methods are reviewed and a taxonomy of existing solutions is presented. Simple geometric error computed directly on the 3D models does not necessarily reflect the perceived visual quality; therefore, integrating perceptual issues for 3D quality assessment is of great significance.

3D mesh models are generally composed of a large set of connected vertices and faces required to be rendered and/or streamed in real time. Using a high number of vertices/faces enables a more detailed representation of a model and possibly increases the visual quality while causing a performance loss because of the increased computations. Therefore, a trade-off often emerges between the visual quality of the graphical models and processing time, which results in a need to judge the quality of 3D graphical content. Several operations in 3D models need quality evaluation. For example, transmission of 3D models in network-based applications requires *3D model compression* and *streaming*, in which a trade-off must be made between the visual quality and the transmission

Figure 2.17: Left: original bunny model; middle: simplified; right: smoothed

speed. Several applications require accurate *level-of-detail (LOD) simplification* of 3D meshes for fast processing and rendering optimization. *Watermarking* of 3D models requires evaluation of quality due to artifacts produced. *Indexing and retrieval of 3D models* require metrics for judging the quality of 3D models that are indexed. Most of these operations cause certain modifications to the 3D shape (see Figure 2.17). For example, compression and watermarking schemes may introduce aliasing or even more complex artifacts; LOD simplification and denoising result in a kind of smoothing of the input mesh and can also produce unwanted sharp features. In order to bring 3D graphics to the masses with a high fidelity, different aspects of the quality of the user experience must be understood.

3D mesh models, as a form of visual media, potentially benefit from well-established 2D image and video assessment methods, such as the Visible Difference Predictor (VDP) [29]. Various metrics have thus been proposed that extend the 2D objective quality assessment techniques to incorporate 3D graphical mechanisms. Several aspects of 3D graphics make them a special case, however. 3D models can be viewed from different viewpoints, thus, depending on the application, view-dependent or view-independent techniques may be needed. In addition, once the models are created, their appearance does not depend only on the geometry but also on the material properties, texture, and lighting [90]. Furthermore, certain operations on the input 3D model, such as simplification, reduce the number of vertices; and this makes it necessary to handle changes in the input model.

The latest advances in visual quality evaluation of 3D graphical models could

be categorized as view-independent and view-dependent metrics. Another approach is applying subjective user tests. The details of each category and a performance comparison of existing metrics are given in the remaining part of this section.

2.4.1 Viewpoint-Independent Quality Assessment

This category of quality assessment metrics directly works on the 3D object space. The quality of a processed (simplified, smoothed, watermarked, etc.) model is generally measured in terms of how "similar" it is to a given original mesh. These similarity metrics measure the impact of the operations on the model. Viewpoint-independent error metrics provide a unique quality value for a model even if it has been rendered from various viewpoints as opposed to the viewpoint-dependent metrics that work on 2D rendered images.

2.4.1.1 Geometric-distance-based metrics

The simplest estimation of how similar two meshes are is provided by the root mean square (RMS) difference:

$$RMS(A,B) = \sqrt{\sum_{i=1}^{n} ||a_i - b_i||^2}, \tag{2.2}$$

where A and B are two meshes with the same connectivity, a_i and b_i are the corresponding vertices of A and B, and $||..||$ is the Euclidean distance between two points. The problem is that this metric is limited to comparing meshes with the same number of vertices and connectivity.

One of the most popular and earliest metrics for comparing a pair of models with different connectivities is the Hausdorff distance [25]. This metric calculates the similarity of two point sets by computing one-sided distances. The one-sided distance $D(A,B)$ of surface A to surface B is computed as follows:

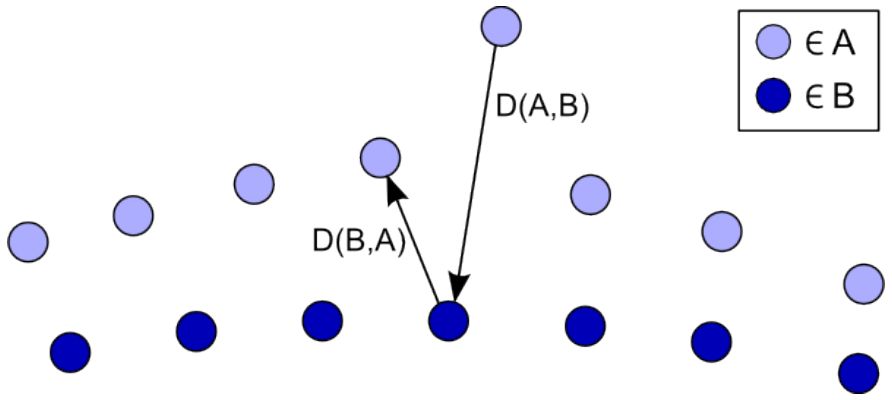


Figure 2.18: The Hausdorff distance between two surfaces. The two-sided Hausdorff distance is $H(A,B) = max(D(A,B), D(B,A))$. (From [19]. ©2011 IEEE, reprinted with permission.)

$$\begin{aligned} dist(a,B) &= min_{b \in B}(||a-b||) \\ D(A,B) &= max_{a \in A}(dist(a,B)). \end{aligned} \tag{2.3}$$

As this distance is non-symmetric, the two-sided Hausdorff distance is computed by taking the maximum of $D(A,B)$ and $D(B,A)$ (Figure 2.18):

$$H(A,B) = max(D(A,B), D(B,A)). \tag{2.4}$$

The Hausdorff distance has been used to find the geometric error between a pair of 3D mesh models in the Metro tool by Cignoni et al. [25]. In this approach the mean distance between a pair of meshes is found by dividing the surface integral of the distance between the two meshes by the area of one of the surfaces. The computation of this integral on discrete 3D models requires a sampling method for fast computation. Aspert et al. [13] also propose a sampling implementation of the Hausdorff distance in the MESH tool.

The Hausdorff distance computes the final distance between two surfaces as the maximum of all pointwise distances. Rather than taking the maximum, extensions have been proposed to provide a better indication of the error across the entire surface. Instead of taking the maximum of the pointwise distances, the average (known as the L_1 norm), the RMS (L_2 norm), and combinations have

been proposed [82, 25].

These metrics are well known and widely used; however, even if they can correctly correlate with human judgement in some simple scenarii, they usually fail to reflect the perceived quality because they compute a pure geometric distance between a pair of meshes, ignoring the working principles of the human visual system. Hence, several other metrics, using different perceptual principles, have been proposed to better estimate the perceived quality of 3D meshes. These solutions can be categorized as roughness-based, structure-based, saliency-based, and strain-energy-based metrics. Since each of these categories focuses on different aspects of perception, it is unlikely for one of them to estimate the perceived visual quality for all scenarii. In this case, blending metrics of several categories may be a possible solution.

2.4.1.2 Roughness-based metrics

Several solutions evaluate the quality of processed 3D models based on their differences from the original model in their surface roughness (or smoothness). These solutions employ the observation that operations on 3D mesh either introduce a kind of noise related to roughness (e.g., as with quantization or watermarking) or cause smoothing of the surface details (e.g., with level-of-detail simplification for rendering). Roughness is an important perceptual property, as we cannot determine the effect of a small distortion if it is on a rough region of the model, and we can detect defects on smooth surfaces more easily. This perceptual attribute, called the *masking effect*, states that one visual pattern can hide the visibility of another.

Karni and Gotsman propose such a roughness-based error metric to evaluate their mesh compression approach [64]. This metric calculates the Geometric Laplacian of a vertex v_i as follows:

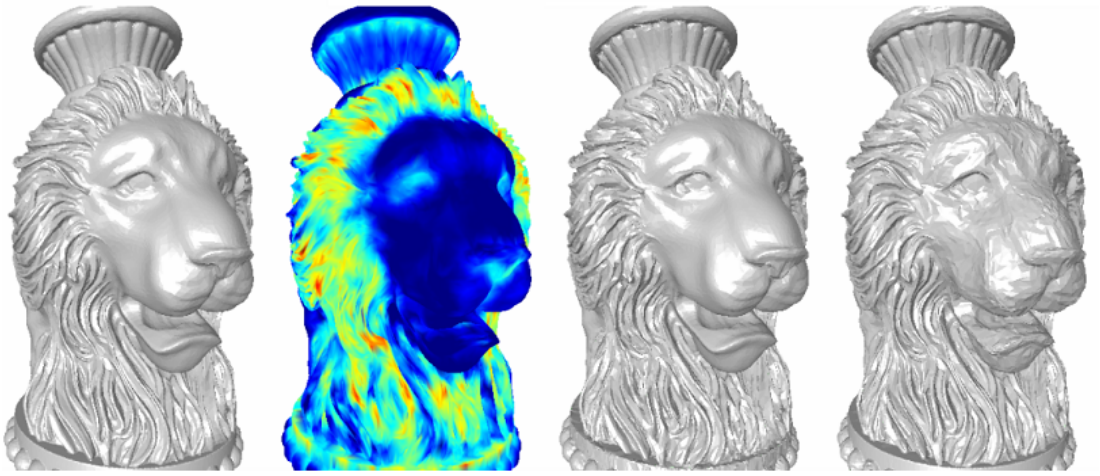


Figure 2.19: Roughness map of a 3D model. From left to right: Original model; roughness map: rough regions shown with warmer colors; noise on rough regions; noise on smooth regions. (From [19]. ©2011 IEEE, reprinted with permission.)

$$GL(v_i) = v_i - \frac{\sum_{j \in n(i)} l_{ij}^{-1} v_j}{\sum_{j \in n(i)} l_{ij}^{-1}}, \tag{2.5}$$

where $n(i)$ is the set of neighbors of vertex i, and l_{ij} is the geometric distance between vertices i and j. Then the norm of the Laplacian difference between models M^1 and M^2 is combined with the norm of the geometric distance between the models as follows (v is the vertex set of M):

$$||M^1 - M^2|| = \frac{1}{2n} \left(||v^1 - v^2|| + ||GL(v^1) - GL(v^2)|| \right). \tag{2.6}$$

One limitation of this metric is that the compared models must have the same connectivity as the RMS error approach.

Wu et al. [129], for driving their simplification algorithm, examine the dihedral angles of the adjacent faces, considering that a rough surface should have greater dihedral angles. Roughness variation has also been used for quality assessment of watermarked meshes; Gelasca et al. [31] and Corsini et al. [28] measure roughness strength by taking the difference between a mesh and its smoothed version. After computing roughness values for the original and watermarked

models, the roughness-based difference is calculated as follows:

$$R(M, M^w) = log\left(\frac{R(M) - R(M^w)}{R(M)} + k\right) - log(k), \tag{2.7}$$

where $R(M)$ is the roughness of the original mesh, $R(M^w)$ is the roughness of the watermarked mesh, and k is a constant to stabilize the numerical results. These roughness-based perceptual metrics [31, 28] have shown to correlate very well with human judgement, particularly in the context of watermarking distortions.

Lavoué proposes a local roughness measure that is able to efficiently differentiate between the different kinds of regions in a mesh: rough parts, smooth regions, and "edge" features, which define border areas between regions [70] (see Figure 2.19). The proposed measure is based on a curvature analysis of local windows of the mesh and is independent of its connectivity. This measure does not estimate any distance but provides a local roughness estimation that can be used to hide artifacts and could be useful for the design of future quality metrics.

2.4.1.3 Structural distortion-based metrics

Structural distortion-based metrics consider the assumption that the human visual system is good at extracting the structural information of a scene in addition to local properties. Lavoué et al. [73] propose Mesh Structural Distortion Measure (MSDM), based on the work of Wang et al. [122], dedicated to 2D images. Instead of extracting the structural information using luminance in 2D images, this metric uses curvature analysis of the mesh geometry. In this work, a local mesh structural distortion measure (LMSDM) on two local windows x and y of the two meshes is calculated as:

$$\begin{aligned} &LMSDM(x, y) = \\ &\quad (\alpha \times L(x, y)^a + \beta \times C(x, y)^a + \gamma \times S(x, y)^a)^{\frac{1}{a}}, \end{aligned} \tag{2.8}$$

with α, β, and γ selected as 0.4, 0.4, and 0.2, respectively, by the authors and with curvature comparison L, contrast comparison C, and structure comparison S computed as:

$$\begin{aligned} L(x,y) &= \frac{||\mu_x - \mu_y||}{max(\mu_x, \mu_y)}, \\ C(x,y) &= \frac{||\sigma_x - \sigma_y||}{max(\sigma_x, \sigma_y)}, \text{ and} \\ S(x,y) &= \frac{||\sigma_x\sigma_y - \sigma_{xy}||}{\sigma_x\sigma_y}, \end{aligned} \tag{2.9}$$

where μ_x, σ_x, and σ_{xy} are respectively the mean, standard deviation, and covariance of the curvature on local windows x and y. Then the MSDM is calculated as follows:

$$MSDM(X,Y) = \left(\frac{1}{n_w} \sum_{i=1}^{n_w} LMSDM(x_i, y_i)^a \right)^{\frac{1}{a}} \in [0,1), \tag{2.10}$$

where X and Y are the compared meshes, x_i and y_i are the corresponding local windows of the meshes, and n_w is the number of local windows. a is selected as 3 by the authors, for equations 2.8 and 2.10 [73]. This metric has proven to correlate very well with human judgement even in difficult scenarii. The authors propose an improved version of this method in [71].

2.4.1.4 Saliency-based metrics

The metrics described above provide a guarantee of the maximum geometric distance rather than estimating the perceived distance between the models.

In this group of metrics the idea is to give more importance to parts of the meshes that gather more human attention. This type of metric is generally used for mesh simplification such that salient parts of a mesh are preserved in the simplification, as suggested by Howlett et al. [57] and Lee et al. [74]. The salient parts of meshes are determined by utilizing an eye-tracker in Howlett et al.'s work, whereas Lee et al.'s method is more convenient as it computes saliency of a mesh automatically, based on its surface curvature.

Similar to the roughness-based and structural distortion-based metrics, saliency uses the perceptual limitation of the human visual system, and its further use for mesh quality assessment is a research area of great interest.

2.4.1.5 Strain-energy-based metrics

Bian et al. [15] propose a solution based on the strain energy on the mesh as a result of elastic deformation. Mesh models are assumed to be elastic objects; as shells composed of triangular faces of negligible thickness. The assumption is that triangle faces do not bend, and each triangle is deformed along its plane by ignoring any rigid body motion.

The perceptual distance between the two versions of the input model is defined as the weighted average strain energy (ASE) over all triangles of the mesh, normalized by the total area of the triangular faces:

$$SFEM(A, B) = \frac{1}{S} \sum w_i W_i, \tag{2.11}$$

where w_i are weights for which several strategies are tested in [15] and W_i is the strain energy computed for triangle i.

This model correlates well with human opinion from the subjective experiment conducted by the authors.

2.4.1.6 Attribute-based metrics

Many 3D mesh models contain per-vertex attributes in addition to the vertex position, such as color, normal, and texture coordinates. Also, in sharp creases of the models, there may be multiple normals per-vertex, or there may be several color values on the boundaries, causing discontinuities in the attributes.

As described by Luebke et al. [82], correspondence between vertices on two surfaces is important but is a difficult issue for meshes with different connectivities; it is difficult to compare attribute values from the original surface and a simplified version in a continuous function. Luebke describes an alternative to Hausdorff, called the bijection method. This requires correspondence between vertices in a 2D parametric domain, such as a texture map. This distance is called a parametric distance. Roy et al. [107] propose a metric called Attribute

Deviation Metric, that can be used to compare two meshes according to their geometric and appearance attributes (or any other per-vertex attributes). The local deviation of attributes between each point of a mesh and the surface of the reference mesh is calculated using parametric distances.

Pan et al. propose a different approach for quality assessment, calculating the quality of a 3D model according to its wireframe and texture resolutions (Equation 2.12) [95].

$$Q(g,t) = \frac{1}{\frac{1}{m+(M-m)t} + \left(\frac{1}{m} - \frac{1}{m+(M-m)t}\right)(1-g)^c} \tag{2.12}$$

Here, m and M are the minimum and maximum bounds of quality, g and t are graphical and texture components scaled into a $[0-1]$ interval, and c is a constant. All coefficients are determined by curve fitting on subjective evaluation data. This metric provides a very good estimation of human judgement as demonstrated in the authors' subjective experiment.

2.4.2 Viewpoint-Dependent Quality Assessment

Viewpoint-dependent quality assessment metrics estimate the perceptual quality of a 3D model as it is shown on the screen; therefore, these metrics are image-based. Viewpoint-dependent metrics can be classified as: *non-perceptual metrics* and *perceptually based metrics*. The visual system does not matter for non-perceptual approaches; they compute the difference between two images pixel by pixel. Perceptually based metrics rely on the mechanisms of the human visual system and attempt to predict the probability that the human observer will be able to notice differences between images.

2.4.2.1 Non-perceptual metrics

Lindstrom and Turk calculate the RMS image error for mesh simplification [77]. In their work, the meshes are rendered from multiple viewpoints and the quality of the resulting luminance images are measured in terms of their differences from the original image as follows:

$$d_{RMS}(Y^0, Y^1) = \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(y_{ij}^0 - y_{ij}^1\right)^2}, \tag{2.13}$$

where Y^0 and Y^1 are m by n luminance images. The RMS metric is not a good metric for image quality assessment and is seldom used because it is highly affected by a shift or scale, and it does not have a perceptual aspect.

Another quality metric for comparing image quality against a reference image consists in calculating the peak signal-to-noise ratio (PSNR). Using the RMS error shown in Eq. 2.13, the PSNR for an image with a highest possible intensity value I_{max} can be calculated by:

$$PSNR = 20log_{10}\left(\frac{I_{max}}{d_{RMS}}\right) \tag{2.14}$$

Although PSNR is also widely used for natural images, it is shown to be a poor indicator of image quality [123]. However, according to a report of the Video Quality Experts Group (VQEG), many more-complicated image quality metrics are not significantly better than PSNR [105]. The reasons for this are discussed in a study of Wang et al. [121].

2.4.2.2 Perceptually based metrics

Many 2D metrics incorporate the mechanisms of the human visual system. These metrics generally use the following perceptual concepts: Contrast Sensitivity Function (CSF), which indicates the relation between the visible spatial frequency

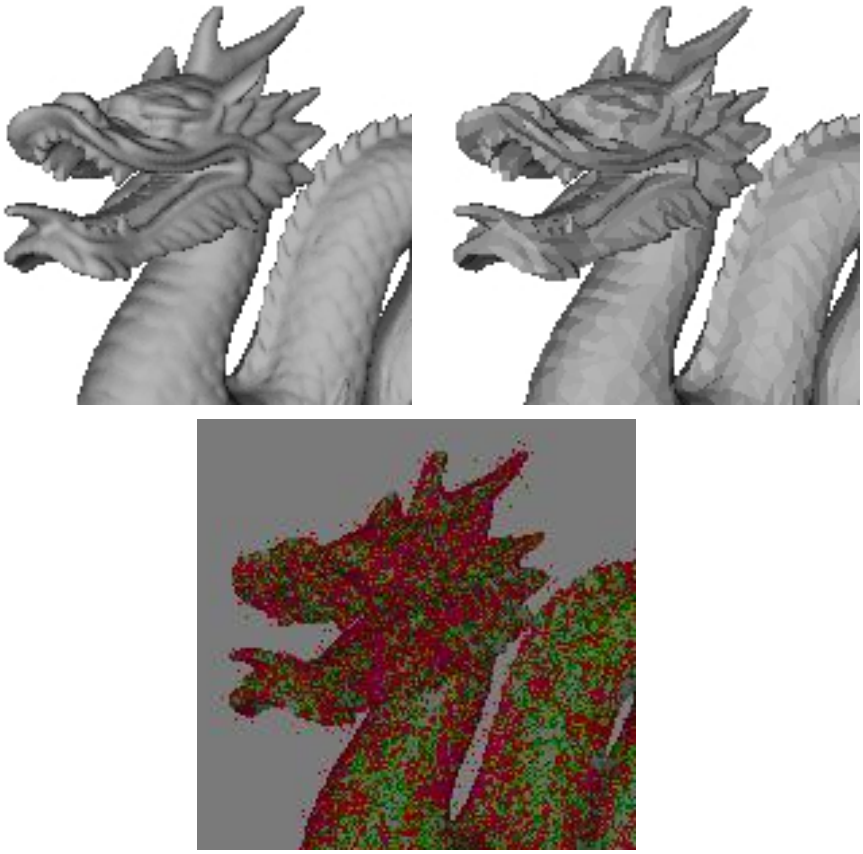


Figure 2.20: Left: original image; right: simplified image; bottom: VDP output. (From [19]. ©2011 IEEE, reprinted with permission.)

and different contrast values; and masking, which describes the reduction in the visual sensitivity of a signal upon the existence of another signal.

A popular metric in this category is Daly's Visible Difference Predictor [29]. This metric takes two images as inputs, one of which is evaluated relative to the other; and the output is an image of the perceptual differences between the two images (see Figure 2.20). The value of each pixel on the output image indicates the detection probability of the difference. The VDP is shown to be a good indicator of perceptually important areas in 3D graphics scenes by the psychophysical experiment of Longhurst and Chalmers [79].

Another well-known metric is the Sarnoff Visual Discrimination Model (VDM) [81] by Lubin. This metric also predicts the detection probability of the differences

between a reference image and the evaluated image, as in VDP. The Sarnoff VDM model works on spatial domain whereas VDP works in the frequency domain; VDM works faster but requires more memory. Li et al. [76] compare the two metrics and find that each model has advantageous properties.

Bolin and Meyer modify the Sarnoff VDM model and propose a simpler and faster metric, which incorporates color properties into their 3D global illumination calculations [18]. This metric is preferred for its efficiency. In their subjective experiment with differently simplified 3D models, Watson et al. [126] show that this metric is an effective predictor of fidelity.

Ramasubramanian et al. [101] propose a perceptually based metric that defines a threshold map in which the minimum detectable difference values are stored for each pixel. This metric handles luminance and spatial processing separately, which provides efficiency since it enables pre-computing of the spatial features.

Ramanarayanan et al. [100] introduce a novel concept, the Visual Equivalence Predictor (VEP), which claims that two images are visually equivalent if they give the same impression even though they have visually different parts. This concept makes more sense for computer-generated imagery in which slightly different illumination techniques lead to different images when analyzed pixelwise although the two images have similar fidelity and information. This model takes 3D geometry, material, and illumination properties into account for the equivalency computations. The VEP concept aims to overcome the limitations of the VDP model, which only considers the earliest levels of visual coding, and is therefore too conservative with respect to the kinds of approximations that can be applied in the rendering process.

Visual masking, which describes the reduction in visual sensitivity of a signal upon the existence of another signal, has been used for view-dependent quality assessment of 3D models. Ferwerda et al. [35] investigate the masking effect for computer graphics and extend the VDP model to include color. In their study, a computational model of the masking effect of the used textures on the artifacts of the 3D meshes is developed. This masking effect is predicted on the varying

contrast, spatial frequency, and orientation features of the texture pattern and on the polygonal tessellation of the model surface.

Three image-quality metrics based on perceptual color differences are proposed by Albin et al. [7]. These similar metrics find the difference between two images in the LLAB (a modified version of CIELAB) color space. The authors state that these metrics are not complete but only initial attempts at a perceptual quality metric. While the first metric is based on a pixel-by-pixel difference of the images, the second metric gives a single distance value using a Monte-Carlo approach, and the last one is a subdivision-based metric, which gives a rougher difference image compared to the first metric in a shorter time.

2.4.3 Subjective Evaluation of 3D Polygonal Models

While automatic metrics are commonly used to predict perceptual quality, relatively few researchers have attempted to measure and predict the visual fidelity of 3D models through subjective experiments. These experiments could be directly used to predict the perceptual quality of 3D models as well as to validate the outcomes of automatic metrics described in the previous sections. Generally, the term "quality" is used to judge how two images (one of them original, the other modified) are "similar" to each other.

2.4.3.1 Experimental measures

Watson et al. study experimental fidelity measures for 3D graphical models [126], and define three of them: *naming time*, which measures the time from the appearance of an object until the observer names it; *rating*, where observers assign a number within a range and meaning determined by the experimenter; and *forced choice preferences*, where observers are shown two or more stimuli, and they choose the stimulus with more of the experimenter-defined quality. The results of this work show that automatic measures of fidelity (e.g., Bolin's [18], Metro [25], mean squared error (MSE) approaches) are successful at predicting

experimental ratings, less successful at predicting preferences, and largely unsuccessful at predicting naming times. On the other hand, when the task is based on comparing different models, ranking is stated to be better than rating the models because the given ratings do not necessarily reflect the perceptual distance between the compared models [104, 111]. The experimental measures used in several user studies can be found in Table 2.1.

2.4.3.2 Experimental design

The parameters used in an experiment are of great importance because they can bias the results significantly, especially for computer-generated stimuli, where almost everything can be controlled. Effective parameters controlled in several quality assessment studies are shown in Table 2.2 and listed as follows:

- *Lighting:* The position and type of light source is a crucial element, with a major effect on the viewing conditions. Rogowitz et al. [104] show that models lit from the front result in different subjective scores compared to the same models lit from above. The human visual system has a prior that light is stationary and comes from a left-above orientation [55].

- *Background:* The background may affect the perceived quality by changing the visibility of the boundaries of the model. While a uniform black background is used in several user studies [126][108], Corsini et al. [28] choose a non-uniform background that fades from blue to white so as not to overestimate the contours.

- *Materials and Shading:* Today, almost all 3D models used in applications have material properties (e.g., texture, normals) and associated complex programmable shaders. On the other hand, most of the subjective evaluations for verifying perceptual metrics do not take material properties into account; they use only diffuse and smooth-shaded models, mostly to prevent highlight effects [28]. Textures have only been used in the context of substituting geometry with texture [95][108]. On the other hand, as described

above, material properties such as textures introduce the masking effect and hide visual artifacts. Researchers often use models without textures or complex material properties to better control the number of variables influencing the outputs.

- *Animation and Interaction:* To evaluate a 3D model in a fair way, observers should be able to see the models from different viewpoints. This can be achieved by animating the object or viewpoint as in [104, 95], as well as giving free viewpoint control to the user as in [28, 111, 72]. Furthermore, animations affect the perception of the models such that, in the study of Rogowitz and Rushmeier [104], artifacts caused by simplification are less visible when the objects are rotating rather than standing still. The sensitivity of the human visual system is dependent on retinal velocity; the eye's tracking ability is limited to 80 deg/sec [30], which should be taken into account when an experiment includes animation.

- *Type of Objects:* There are several concerns to keep in mind when selecting objects for a subjective experiment. Watson et al. [126] state that evaluation results are different for animal models and man-made artifacts. Further, using abstract objects helps avoid semantic interpretation [108]. Also, the complexity and roughness of the models are important. In a very complex object, simplifications may not be visible and the roughness of a mesh may mask artifacts.

- *Masking:* The object's geometry, roughness, texture, and applied noise or watermarking can mask each other. Lavoue et al. [70] examine the masking effect of noise and roughness; Pan et al. [95] and Rushmeier et al. [108] examine the masking effect of textures on geometry. The masking effect should be considered while designing an experiment.

- *Extent:* The extent, i.e., the display area of the rendered model in pixels, should be large enough to reflect the details of the model. Showing too many items simultaneously may decrease the visibility of the models. The display extents used in several user studies can be found in Table 2.2.

- *Levels:* When an operation (simplification, watermarking, etc.) on meshes

is to be tested, the number of the comparison cases and the strengths of the applied operations for each case should be adjusted carefully. Too few levels (compared cases) may not sufficiently reflect the tested operation, whereas a large number of levels may not be feasible, as they would require too many subjects. For simplification case, there are studies using three [126, 104] to seven [111] levels (including the originals) of simplification.

- *Stimuli order:* In comparison-based experiments, stimuli can be shown to the user simultaneously (e.g., side-by-side) or in succession (e.g., first the reference, then the tested models). When they are shown in succession, enabling users to turn back to the reference model as in the experiment of Rogowitz and Rushmeier [104], allows for a more-detailed comparison. Also, the order and the position of the stimuli should be selected in a way that minimizes the effect of external variables such as observer movements and room's ambient light.

- *Duration:* The duration of which the tested models are shown to the subjects may also affect the results of evaluation.

2.4.3.3 Standards for subjective evaluation

Although no specific recommendation for subjective evaluation of 3D models exists currently, a number of standards, which define the conditions for subjective experiments for other multimedia content (e.g., image and video), could be adapted and used. A well-known standard is the ITU-R BT.500 Recommendation [2], which defines the methodology for the subjective evaluation of image quality. Different experiment methods, such as double-stimulus continuous quality-scale (DSCQS) and simultaneous double stimulus for continuous evaluation (SDSCE) are recommended and grading scales and how to present test materials are outlined. Several of these methods, which may be useful for quality assessment of 3D meshes, are briefly explained below.

- The DSCQS method is recommended for measuring the relative quality of a system against a reference. It has a continuous grade scale which is

	Masking	Task	Measures	Levels	Purpose of test
Watson01		mesh simp.	rating, preference, naming time	3 (20%, 50%, orig)	
Rogowitz01		mesh simp.	rating	3 (25%, 40%, orig)	to evaluate still images for geo. models
Corsini07		watermarking	rating	E1: 4 mod x 3 wm levels x 3 res. + 4 orig E2: 4 mod x 11 wm types + 4 orig	to fit a metric
Silva08		qual. assesment on mesh simp.	ranking 4 models	10, 20, 27, 35, 43, 50%	to calculate quality of simplified meshes
Pan05	texture-mesh geometry	fitting of subject eval. results to perceptual metric	rating – 5 levels	5 objects, 6 levels of mesh resolution x 3 levels of texture resolution	to fit a metric
Lavoué10	noise-roughness	compression, watermarking, smoothing	rating	E1: orig + 3 levels of noise E2: 4 orig + 4x3 noise on smooth + 4x3 noise on rough E3: 4 orig + 4x9 smooth + 4x12 noise versions	to compare different objective metrics
Rushmeier00	texture-mesh geometry	Simplification	rating: 0...100	2 objects x 3 geometry levels (full; 47x reduction in size; 94x reduction in size) x 4 texture levels (none, 512x512, 256x256, 64x64)	to examine possibility of substituting texture for geometry

Table 2.1: Experiment methodologies of recent subjective experiments on quality assessment. (From [19]. ©2011 IEEE, reprinted with permission.)

	Lighting	Animation/ Interaction	Materials	Background	Object	Extent of Stimulus	simultaneous / successive
Watson01 [126]	oblique	no / no	no	black	man-made vs. animal	591px in width	simultaneous
Rogowitz01 [104]	above, colocated with view	rotating object / no	no		simple vs. complex (vertex count)		successive, can go back
Corsini07 [28]	white point light, top corner of obj. bbox	no / free interaction	diffuse only	blue to white fading		600x600 px	
Silva08 [111]		no / free interaction		black	have small num. of vertices and different nature		simultaneous
Pan05 [95]	front	rotation / adjustable rot. speed	textured objects			750px in height	simultaneous
Lavoué10 E1, E2	front	no / free interaction	diffuse only	black	objects with smooth and rough regions		simultaneous
Lavoué10 E3 [72]	white point light, top corner of obj. bb.	no / free interaction	diffuse only	non-uniform	objects from different natures		successive
Rushmeier00 [108]	above, colocated with view	no / no	textured objects	black	abstract objects	370px in height	

Table 2.2: Experiment design of recent subjective experiments on quality assessment. (From [19]. ©2011 IEEE, reprinted with permission.)

partitioned into five divisions of equal length, labeled *bad*, *poor*, *fair*, *good*, and *excellent*. Subjects can mark the scale in a continuous manner and then the grades are mapped to a 0-100 interval. The reference and test material are shown twice in succession.

- The SDSCE method is recommended for measuring the fidelity between two impaired video sequences. The stimuli are shown side by side and the grading is continuous.
- The ITU-R BT.500 standard also includes recommendations related to the evaluation of the experiments, such as how to eliminate the outlier data.

A related standard, the ITU-T P.910 recommendation [1], describes subjective assessment methods for evaluating the one-way overall quality for multimedia applications. This recommendation addresses test methods and experiment design, including comparison methods; and evaluation procedures, including viewing conditions and the characteristics of the source sequences, such as duration, kind of content, number of sequences, etc. Subjective evaluation of 3D graphical models as a form of media can benefit from these recommendations.

Chapter 3

Visual Attention Models

When searching for visually attractive points in a 3D scene, we have two types of concerns. One of the concerns is, among many objects, which objects stand out and capture most of the viewers' attention. In addition to identifying the visually significant objects in a scene, finding out the visually attractive parts/regions of these objects is another concern. In this chapter, we handle these concerns separately, and present two new saliency computation models for different types of graphical contents.

The first model, Per-Vertex Saliency (PVS) model works on individual and possibly animated 3D mesh models and finds saliency values for each vertex of the input meshes, whereas the second model, Per-Object Saliency (POS) works on separate objects and finds saliency values for each object in the scene according to their motions. Additionally, we present Extended Per-Vertex Saliency (EPVS) model which is an extension of PVS and makes use of the human visual system principles used in the POS model when calculating motion based saliencies.

3.1 Per-Vertex Saliency Model

The metric is targeted for calculating the saliency of 3D meshes, and it is applicable to not only static but also vertex-animated models. Most of the previous saliency detection models in the literature [74, 78, 34] focus on the shape and curvature features of 3D meshes. On the other hand, when applied to generic 3D models, they all suffer from problems caused by their omission of material properties or complexity introduced by animation. Besides, curvature-based methods are only capable of detecting salient locations with distinct curvature features, which are insufficient for our animation-based saliency computation task.

The combination of different features, such as color, orientation, spatial frequency, brightness, direction of movement, into a single saliency model requires an integrated model of attention. The feature-integration theory of attention [117], which has been used successfully for 2D images, suggests that the visual scene is initially coded along a number of separable dimensions; and the contribution of any features which are present in the same region are combined. Based on this theory, Itti et al. have proposed a model for integration of the different features in 2D images [62]. Our model of saliency for 3D animated models uses a similar approach.

In order to derive an integrated approach for saliency calculation, our metric considers multiple features of animated meshes such as their color, geometry, and motion; each as a separate channel. The general structure of the proposed approach is shown in Figure 3.1. The 3D model is decomposed into a set of dimensions, with each dimension stored in a separate feature map. Different regions of the 3D model then compete for saliency within each feature map, and only those regions that stand out locally from their neighborhood in different scales are kept. Then, the saliency values computed for each dimension are combined into a master saliency map, which approximates the overall attended regions of the animated 3D mesh.

Our saliency calculation depends on the center-surround mechanism of the

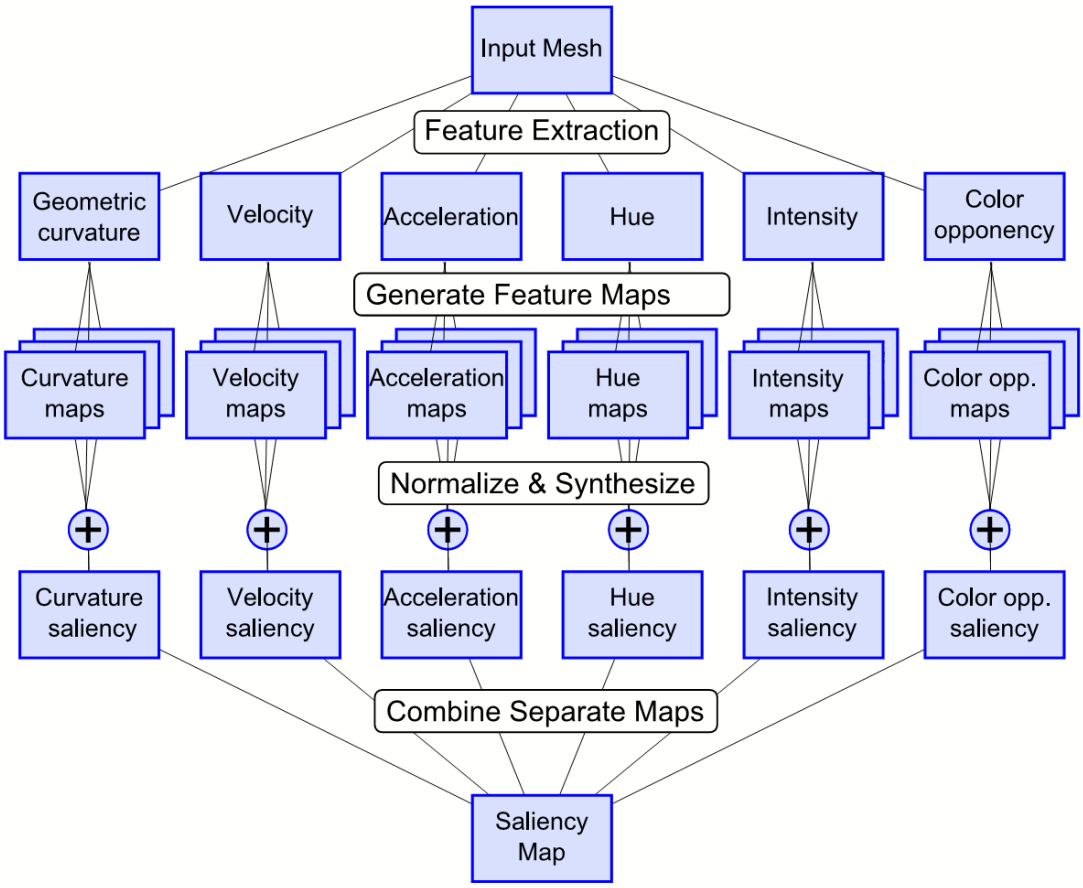


Figure 3.1: The proposed saliency computation framework. (From [21]. ©2010 ACM, reprinted with permission.)

human visual system [62] [74]. The mechanism captures the regions that are spatially different from their surroundings. For each feature, we compute the salient vertices by a set of center-surround operations. In these operations, vertices in a small neighborhood of a vertex v constitute the center, and vertices in a larger neighborhood constitute the surround. The across-scale difference of a feature between the central region and the surrounding region determines the saliency.

We compute different feature maps of multiple scales to account for the saliency in different scales of the mesh. For example, a small scale saliency map may detect the movement of a finger in a human model but it will fail to detect a larger-scale movement such as that of a leg. A large-scale saliency map will fail in the former case and succeed to show the saliency correctly in the latter case.

As shown in Figure 3.1, our saliency computation framework consists of four steps: (i) feature extraction, (ii) generating feature maps, (iii) normalization and synthesis, and (iv) combining separete saliency maps.

3.1.1 Feature Extraction

Different features of objects in a visual scene affect the direction of our attention. That is, a region in a visual scene could stand out due to different features. For example, in a scene with similar colors at all regions, presence of a differently colored object results color properties to be the dominant feature to direct our attention; however, if number of differently colored objects increases, another feature channel, for instance motion, could be the the most significant channel. Since the visual attractiveness could be directed by various properties of objects, considering more feature channels increases the chance of finding out the most salient object. We use geometry, velocity, acceleration, hue, color opponency, and intensity features of a vertex in our saliency computations. Each feature is calculated as follows for each vertex:

Geometry: This feature is used for computing the saliency of a 3D mesh due to its shape. Curvature is a significant feature of a vertex that can indicate

its distinctiveness among others. Therefore, for geometry-based computations, we use the mean curvatures of vertices, as previously proposed by Lee et al. [74]. For computing mean curvatures, we use Meyer et al.'s method for curvature computation [88].

Velocity: The velocity of a vertex is calculated by taking the positional difference of a vertex in consecutive frames as follows:

$$vel(v_i, f_i) = \frac{p(v_i, f_j) - p(v_i, f_{j-1})}{diagonal}, \tag{3.1}$$

where v_i stands for vertex i; p stands for position which is a vector of length three; f_j stands for j^{th} frame; *diagonal* is the length of the diagonal of the mesh's axis-aligned bounding box (Figure 3.2). The division by *diagonal* makes our method scale independent.

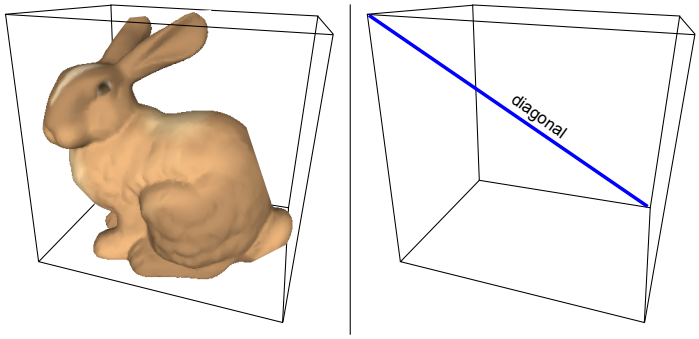


Figure 3.2: Left: Axis-aligned bounding box; right: diagonal of the axis-aligned bounding box.

Acceleration: The acceleration computation is very similar to the velocity calculation. It is calculated by taking the difference of velocities on consecutive frames but this time diagonal is not used because its effect is already present in velocities.

$$acc(v_i, f_i) = vel(v_i, f_j) - vel(v_i, f_{j-1}), \tag{3.2}$$

where $acc(v_i, f_i)$ stands for acceleration of vertex i at frame j.

Hue: We extract hue values from the RGB color of a vertex and map them

to the 0-360 interval. Let r, g, and b denote the red, green and blue components of the color of a vertex. Corresponding hue value is extracted as follows.

$$hue = \begin{cases} 0, & \text{if } max = min \\ \left(\frac{60\times(g-b)}{max-min} + 360\right) \bmod 360, & \text{if } max = r \\ \left(\frac{60\times(b-r)}{max-min} + 120\right) \bmod 360, & \text{if } max = g \\ \left(\frac{60\times(r-g)}{max-min} + 240\right) \bmod 360, & \text{if } max = b \end{cases} \tag{3.3}$$

where max and min stand for maximum and minimum values among r, g, and b.

Because the hue values wrap around (e.g., the value 359 is close to 1), while computing the center-surround differences (explained in Section 3.1.2) we use the smaller distance between two values. For example, the distance between values 350 and 10 is 20 instead of 340.

Color opponency: In color perception, there is an opponent color theory proposed by Hering [48]. This theory states that human visual system interprets red-green and blue-yellow colors in an opposite fashion so that we cannot perceive a color such as yellowish blue or redish green. For the central area of the visual field, the neurons in the primary visual cortex are excited by one of the colors in the Red-Green and Blue-Yellow pairs, and inhibited by the other color. The opposite holds for the surrounding area. Therefore, presence of a color in the center increases saliency of a region, if this region is surrounded by the opponent color. We use the color opponency values described in [62]. For color opponency, the feature values for center and surround are computed separately as follows:

$$\begin{aligned} O_{RGC} &= max((r - \tfrac{g+b}{2}), 0) - max((g - \tfrac{r+b}{2}), 0) \\ O_{RGS} &= max((g - \tfrac{r+b}{2}), 0) - max((r - \tfrac{g+b}{2}), 0) \\ O_{BYC} &= max((b - \tfrac{r+g}{2}), 0) - max((\tfrac{r+g}{2} - |\tfrac{r-g}{2}| - b), 0) \\ O_{BYS} &= max((\tfrac{r+g}{2} - |\tfrac{r-g}{2}| - b), 0) - max((b - \tfrac{r+g}{2}), 0) \end{aligned} \tag{3.4}$$

Intensity: For each vertex, we compute the intensity feature by calculating the average of the RGB components of the vertex.

The calculated per-vertex features are used to generate feature maps for each individual feature which is presented in the following part. These feature maps are not actually 2D maps but vector's holding values of each specific feature for each vertex. Separating these features have the advantage of identifying each feature's individual saliency impact and enables tuning each feature's weight on generating the final saliency map.

3.1.2 Generating Feature Maps

After processing all features of the vertices, we calculate the Gaussian-weighted center-surround differences for several center-surround scales, and then we generate a separate feature map for each pair of center-surround scale and feature. Define this feature map as $featuremap(c, s, f)$ which stores the feature for each vertex of a 3D mesh, where c and s stand for the center and surround levels and f stands for the feature. For scale (c, s) and feature f the entry for vertex i in the map is generated as follows:

- Let the neigborhood $N(v, d)$ of vertex v be the set of vertices that have a Euclidean distance smaller than d to vertex v. First, we calculate the Gaussian-weighted average of feature f for the vertices that are in $N(v_i, 2c)$ [62].

$$G(f, c, v_i) = \frac{\sum_{x \in N(v_i, 2c)} f_x exp\left(-\frac{||v_x - v_i||^2}{2c^2}\right)}{\sum_{x \in N(v_i, 2c)} exp\left(-\frac{||v_x - v_i||^2}{2c^2}\right)} \tag{3.5}$$

- Then, the absolute center-surround differences are stored in the feature map.

$$featuremap(c, s, f, i) = |G(f, c, v_i) - G(f, s, v_i)|, \tag{3.6}$$

  where c is the fine (center) scale and s is the coarse (surround) scale.

For center and surround distances, Itti et al. [62] have used 2, 3, and 4 pixels for fine scale c and $c + \delta$ pixels for coarse scale s, where $\delta \in \{2, 3\}$, resulting in six different center-surround levels. On the other hand, Lee et al. [74] used 2ϵ,

3ϵ, 4ϵ, 5ϵ, 6ϵ, where ϵ is 0.3% of the diagonal of the bounding box for the center and the surround is the double of the center. In this case, five different levels of center-surround scales are considered. The largest surround scale of the second approach covers only $12 \times 0.3\% = 3.6\%$ of the diagonal of the bounding box, which is not large enough since, for example, in a human model it approximately corresponds to a finger size and cannot reflect a center-surround difference for a larger region. Hence, we also need to consider larger center-surround scales as well as the narrow scales. The list of center-surround levels used for our calculations follows:

$$\begin{aligned} &L = \{2\epsilon, 3\epsilon, 5\epsilon, 8\epsilon, 13\epsilon, 21\epsilon, 34\epsilon, 55\epsilon\}, \\ &(c, s) \in \{x, y | x = L[i] \wedge (y = L[i+1] \vee y = L[i+2])\} \end{aligned} \tag{3.7}$$

where c stands for center, s stands for surround and ϵ again means 0.3% of the diagonal. Above equation means that for each central neighborhood, two surround levels exist which are the smallest two neighborhoods greater than the central neighborhood. For instance, central neighborhood 2ϵ is used for two center-surround levels, $(2\epsilon - 3\epsilon)$ and $(2\epsilon - 5\epsilon)$. Using the Fibonacci sequence decreases the cost of calculating neigborhoods of different scales such that only eight different neighborhoods are calculated to get thirteen different center-surround levels. In this case, the largest surround level covers $55 \times 0.3\% = 16.5\%$ of the bounding box.

3.1.3 Normalization of Feature Maps

After calculating the separate feature maps, the next step of our method is to combine the maps that belong to the same feature. This is done by linear addition after normalizing the maps using the normalization method defined by Itti et al. for 2D images [62]. For example, all feature maps related to velocity are normalized and summed up to determine the velocity-based saliency map. This normalization method works as follows:

- All values in the feature map are mapped to a fixed range $0 - M$ so that the maximum saliency becomes M.
- All vertices with a saliency greater than all of its neighbors' saliencies are signed as local maximums. Let a be the average value of the saliencies of the local maximums.
- The feature map is multiplied with $|M - a|^2$.

Using this normalization technique suppresses the maps in which the saliency values are distributed homogeneously, whereas a map with an outstandingly salient point is promoted. After normalization, all maps related to a feature are linearly added and we get a saliency map for each feature.

Other normalization methods, such as iterative competition between salient locations and simple normalized summation, have been proposed by Itti and Koch for 2D images [59]. For 3D animated models, however, this approach is the most suitable one regarding the computation-accuracy tradeoff. In a 3D animated mesh, several frames have to be processed each having a large number of vertices, which makes the iterative competition solution computationally indesirable where the normalization method we use works faster and provides comparable quality.

After computing the saliency maps based on each feature, we combine these maps to a final saliency map. This combination is performed by linear addition as each feature map has already been normalized in the previous step.

3.1.4 Results

In this section, we demonstrate the results of our saliency metric. Each feature has different contribution to the saliency calculation; therefore, it is useful to show the effect of each feature separately before demonstrating the final saliency map.

Geometry based saliency map: It identifies the important regions of a 3D mesh according to its shape. It can be considered as the static saliency of a 3D

mesh. Figure 3.3-a shows salient parts of a horse model due to its geometry. As seen in the figure, the parts of the mesh that are outstanding according to their mean curvatures (e.g., eyes, feet, joint of tail) are computed as salient.

Velocity and acceleration based saliency maps: Velocity and acceleration features are used to obtain the regions that are salient due to their motion. Note that high saliency of a region is not the direct result of high velocity or acceleration of this region. Saliency is related to the difference of this region's motion with respect to the surrounding area.

Figures 3.3-b and 3.3-c show examples of velocity-based and acceleration-based saliency maps, respectively. Although these two features are similar, each one is sensitive to a different behaviour of a motion. For example, while velocity-based saliency map finds left feet as salient (Figure 3.3-b), the acceleration-based saliency map finds the front-right foot of the horse model as more salient (Figure 3.3-c). In Figure 3.3-d, the combination of motion related saliency maps are shown.

Hue, color opponency, and intensity based saliency maps: In addition to the shape and motion related attributes of 3D meshes, our saliency metric also considers per-vertex material properties. Figure 3.4-b shows the hue-based saliency map of the cloth model shown in Figure 3.4-a. As seen in this figure, the hue-based saliency map highlights the regions that have different hue values than their neighbors. Color opponency-based saliency map identifies the regions that are surrounded by the opponent color. In Figure 3.4-c, we can see that the green parts are indicated as salient since in the original image these regions are surrounded by the opposite color (red). Another color related attribute which is used to identify the salient regions is the intensity of color. The intensity-based saliency map, which is shown in Figure 3.4-d, points out the salient regions due to their color intensity.

Final saliency map: All features are combined to obtain a final saliency map. In Figure 3.5, example saliency maps belonging to different animated models are shown. Although these images are only snapshots of the animations of the models, looking at the saliency images we can understand the salient regions of

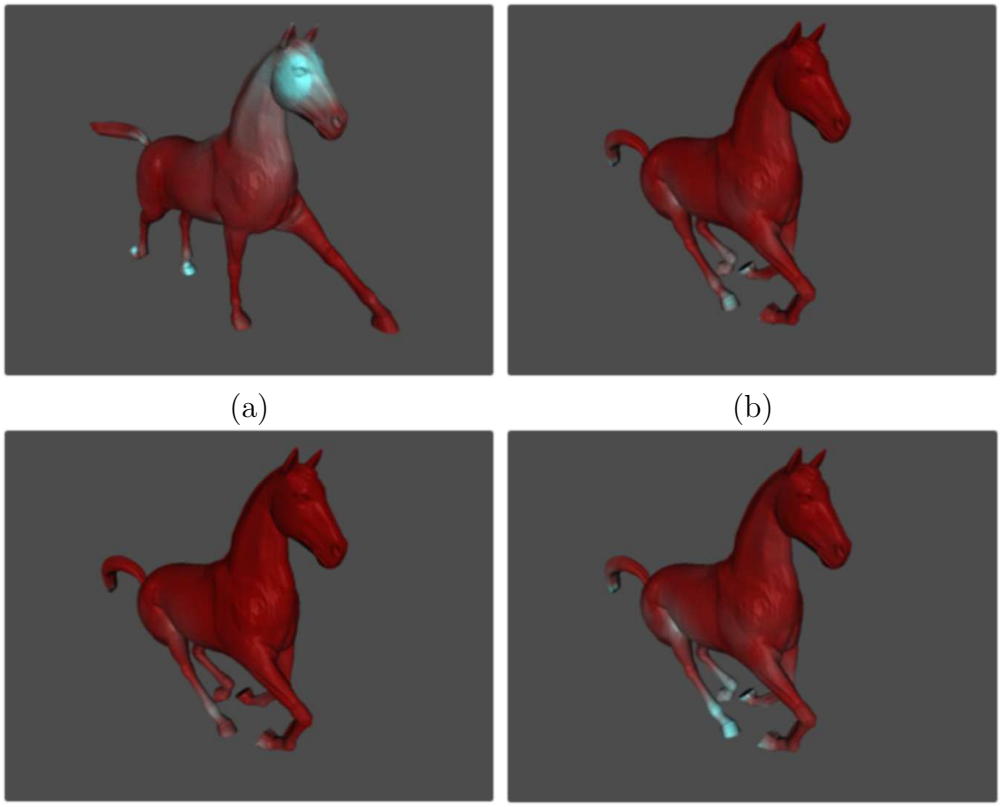


Figure 3.3: The calculated saliencies based on geometric mean curvature (a), velocity (b), and acceleration (c) in a horse model. The image in (d) shows the combined saliency map of the velocity and acceleration features. Light-colored areas show the salient regions and are emphasized for illustration purposes. (From [21]. ©2010 ACM, reprinted with permission.)

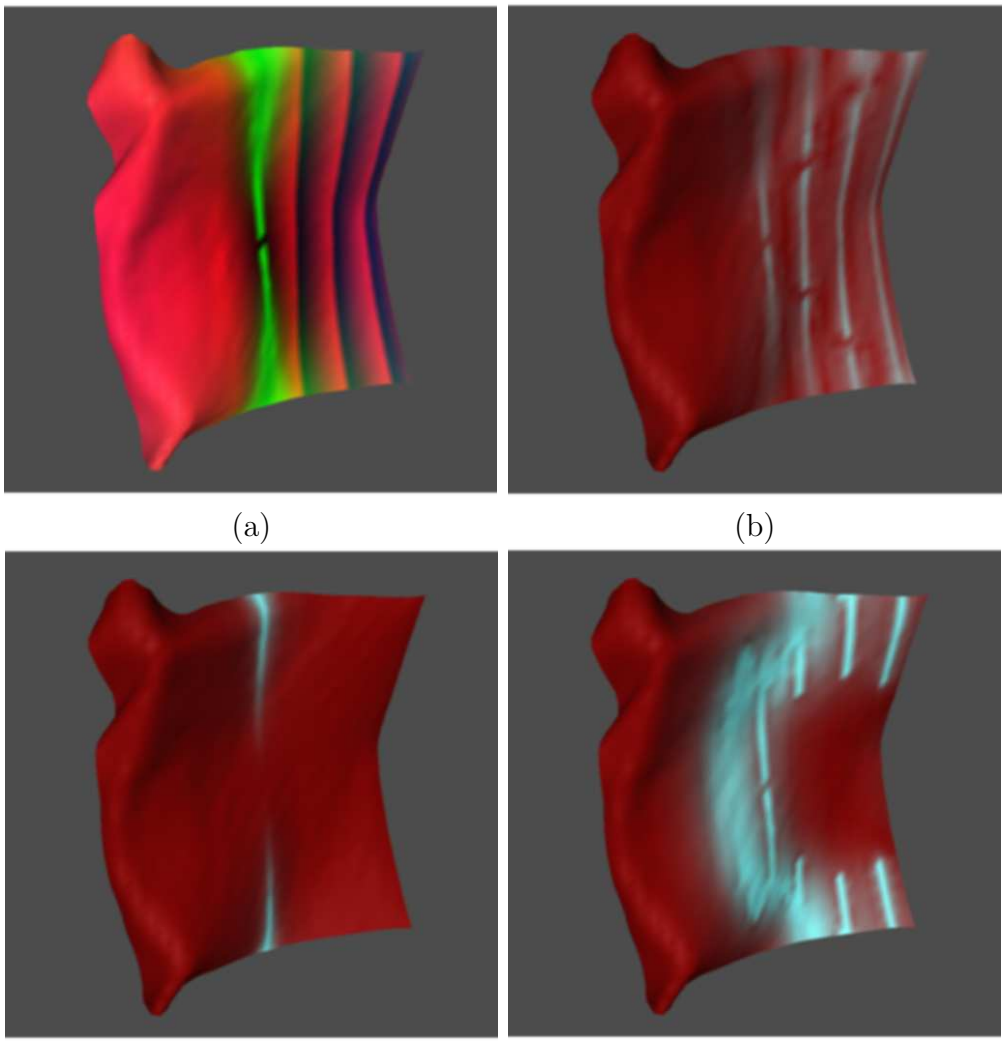


Figure 3.4: The animated cloth model (a). The calculated saliencies based on hue, color opponency, and intensity are shown in (b), (c), and (d), respectively. Light-colored areas show the salient regions and are emphasized for illustration purposes. (From [21]. ©2010 ACM, reprinted with permission.)

the models due to their animations, such as the legs of the horse model and the left part of the cloth model.

3.1.5 Applications

The computed saliency map can be used for different applications, including mesh simplification, viewpoint-selection [74], persuading attention [67], and accelerating global illumination computations [134].

Our saliency calculation is performed in a view independent way. Once the saliency values are computed as a preprocessing step, they can be stored as per-vertex attributes in a 3D animated mesh and can be used in different applications without recomputing. We present several applications in which the usage of saliency has a significant importance.

3.1.5.1 Mesh Simplification

The first application in which the usage of saliency is illustrated is simplification of 3D animated meshes. This application is based on the idea that a vertex with a higher saliency value means that it resides on a perceptually more interesting region of the mesh. Our goal is to delay the simplification of the salient parts of the mesh, because those parts are presumably the parts of the mesh where the viewers focus on. On the other hand, saliency maps themselves do not suffice to be used as the main simplification metric, but they are rather used as heuristics for simplification.

We have used the Quadric Error Metric (QEM) described in [40] as our main simplification metric and incorporated saliency maps as a supporting decision factor. QEM is based on iterative pair contractions. A pair contraction is uniting two vertices into a single vertex in a way that minimizes the resulting geometric error. In QEM, All possible edge pairs to contract are kept in a heap according to their costs (geometric errors). The edge pair with the minimum cost is contracted to a single vertex and the heap is updated after contraction.

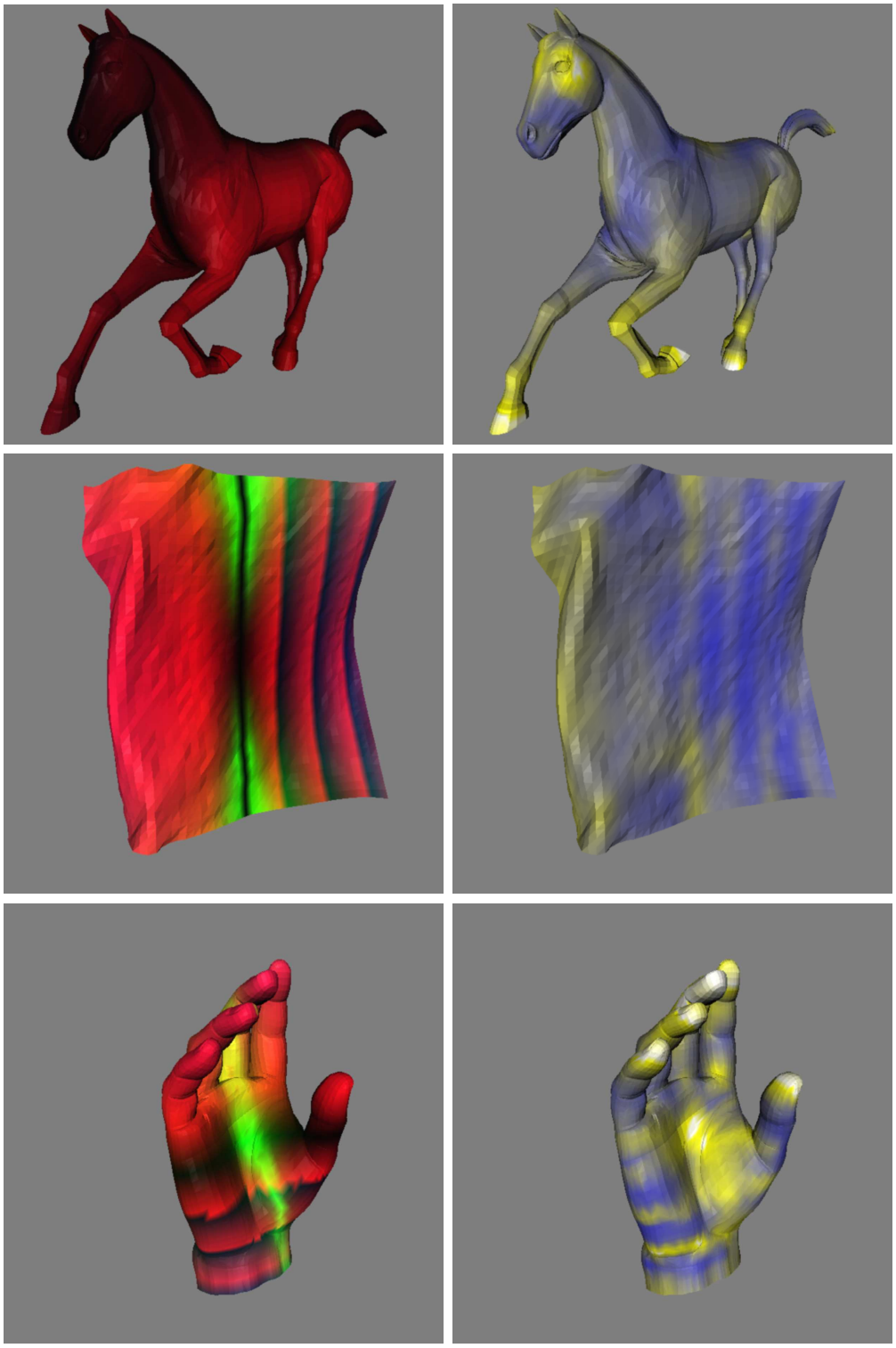


Figure 3.5: Left: The models with their original views, right: the final saliency maps of these models. (From [21]. ©2010 ACM, reprinted with permission.)

In our framework, the QEM scores, determining the simplification possibilities of vertex pairs are blended with the saliency value of the vertices and the final score is assigned to the vertices, according to the following equation:

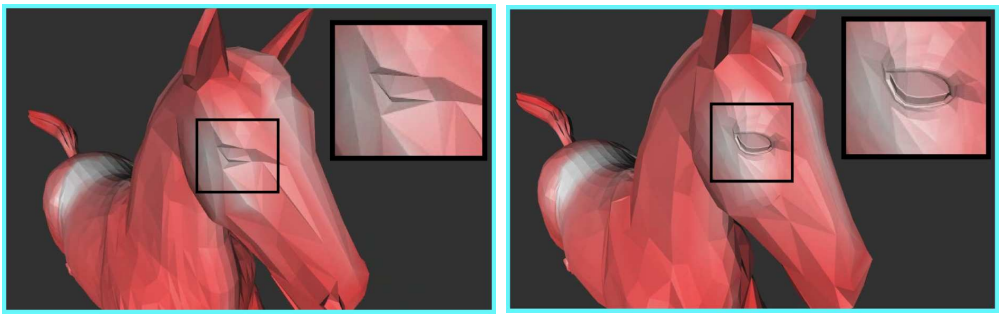
$$w_i = -q_i \times (1 - \alpha) + s_i \times \alpha \times \mu, \tag{3.8}$$

where w_i is the final weight, q_i is the QEM score, s_i is the saliency score of pair i, α is the weight of the saliency score in simplification and μ is the normalization factor of the saliency values.

In our experiments, we have selected $\alpha = 0.5$ and $\mu = 10^{-4}$. After the scores are assigned to each vertex, the vertices are placed on a heap, i.e., a priority queue, according to their scores. A vertex with a high score means that it has a lower QEM error score and/or a higher saliency; thus, we delay the contraction of the vertices with higher scores. For this purpose, we use a min-heap where vertices with lower scores are placed at the top. These vertices are consequently simplified earlier without degrading the quality of the salient regions of the mesh and without increasing the QEM error rate significantly. When two vertices are contracted to a single wertex, we assign a new score to the resulting vertex. This score is obtained by the own QEM score of the new vertex and a saliency value which is computed by linearly interpolating the saliency values of the contracted vertices. This interpolation is performed depending on the distance of the new vertex to the contracted vertices. Figure 3.6 shows a frame of the result of our simplification method applied on the horse model. The figure illustrates that the new method successfully preserves the salient regions such as the eye of the horse model, and the colored region on the back.

3.1.5.2 Dynamic Level-of-Detail

Dynamically adjusting the level-of-detail of the models in the scene depending on the display area they allocate is a widely used technique. This technique provides rendering efficiency by avoiding the time to render the imperceptible details of complex meshes. The view-independent saliency values that are computed as a preprocess can be used for real-time level-of-detail adjustment. The idea here is



(a) (b)

Figure 3.6: The animated horse model is simplified using quadric error metrics (a) and using our saliency-based simplification method (b). Both of the models have 4600 faces. (From [21]. ©2010 ACM, reprinted with permission.)

that the viewers mostly focus on objects that are closer to them in the scene, or the objects that allocate the most space on the screen. Therefore, we can further simplify objects that are further away from the viewer. This simplification is performed considering the saliency values of the vertices as explained in Section 3.1.5.1.

Another usage area of saliency based mesh simplification is the dynamic level-of-detail adjustment for 3D animated models. Depending on the display area allocated by the models, the view independent saliency values that are computed as a pre-process can be used for real-time level-of-detail adjustment.

3.1.5.3 Viewpoint Selection

The saliency information can also be used to automatically determine the viewpoint for an animation. Since the saliency values indicate the significant regions of 3D meshes, automatic selection of the viewpoint through animation would be a useful tool for a number of applications. A similar approach can be used for automatically creating a thumbnail for animated 3D meshes; however, in this case, it would be better to use the saliency that is generated using only the geometric and the material properties of a 3D mesh without the motion related attributes since a thumbnail does not include animation. Alternatively, the frame with the largest saliency can be selected.

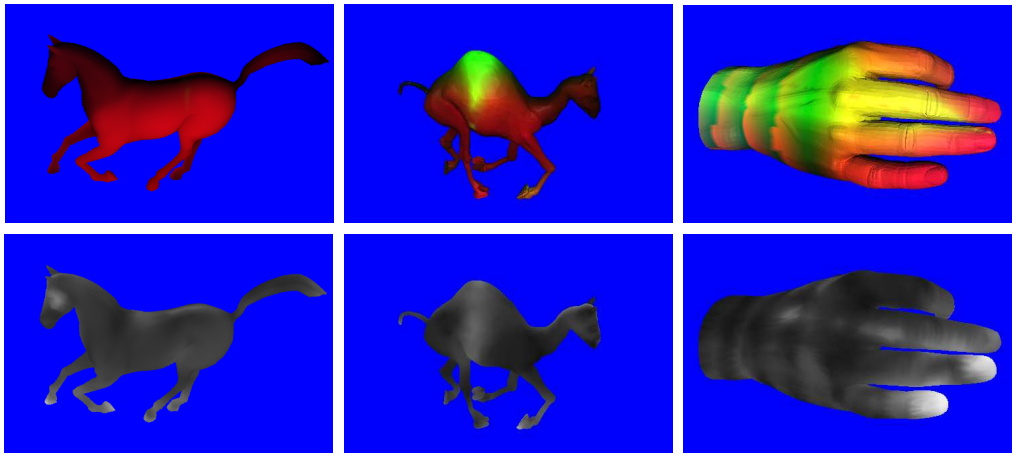


Figure 3.7: Selected viewpoints for several meshes. **top:** original views of the models, **bottom:** corresponding saliency maps. (From [21]. ©2010 ACM, reprinted with permission.)

In order to select the viewpoint automatically, we examine a spherical region around a 3D mesh and the viewpoint in which the total saliency of all visible points reaches the maximum is selected as the viewpoint. Figure 3.7 shows the selected viewpoints for several animated models. This technique can further be extended for automatic control of camera position in an animation.

3.1.5.4 Quality Assessment Metrics and Saliency Guided Simplification

In this section, we compare the saliency guided simplification (SGS) method proposed in Section 3.1.5.1 with Garland's quadric error based simplification method (QEM). In our analysis we generated three levels of simplified meshes. The first level is the original mesh (reference mesh). The number of vertices in the meshes of second and third levels are 1/2 and 1/4 of the reference mesh, respectively. The models and their simplified versions are shown in Figures 3.8 and 3.9. In the figures, it is apparent that the models' details are preserved on the salient regions of meshes. For example, eyes of the horse and nails of the finger are better preserved with the SGS method, compared to the QEM method.

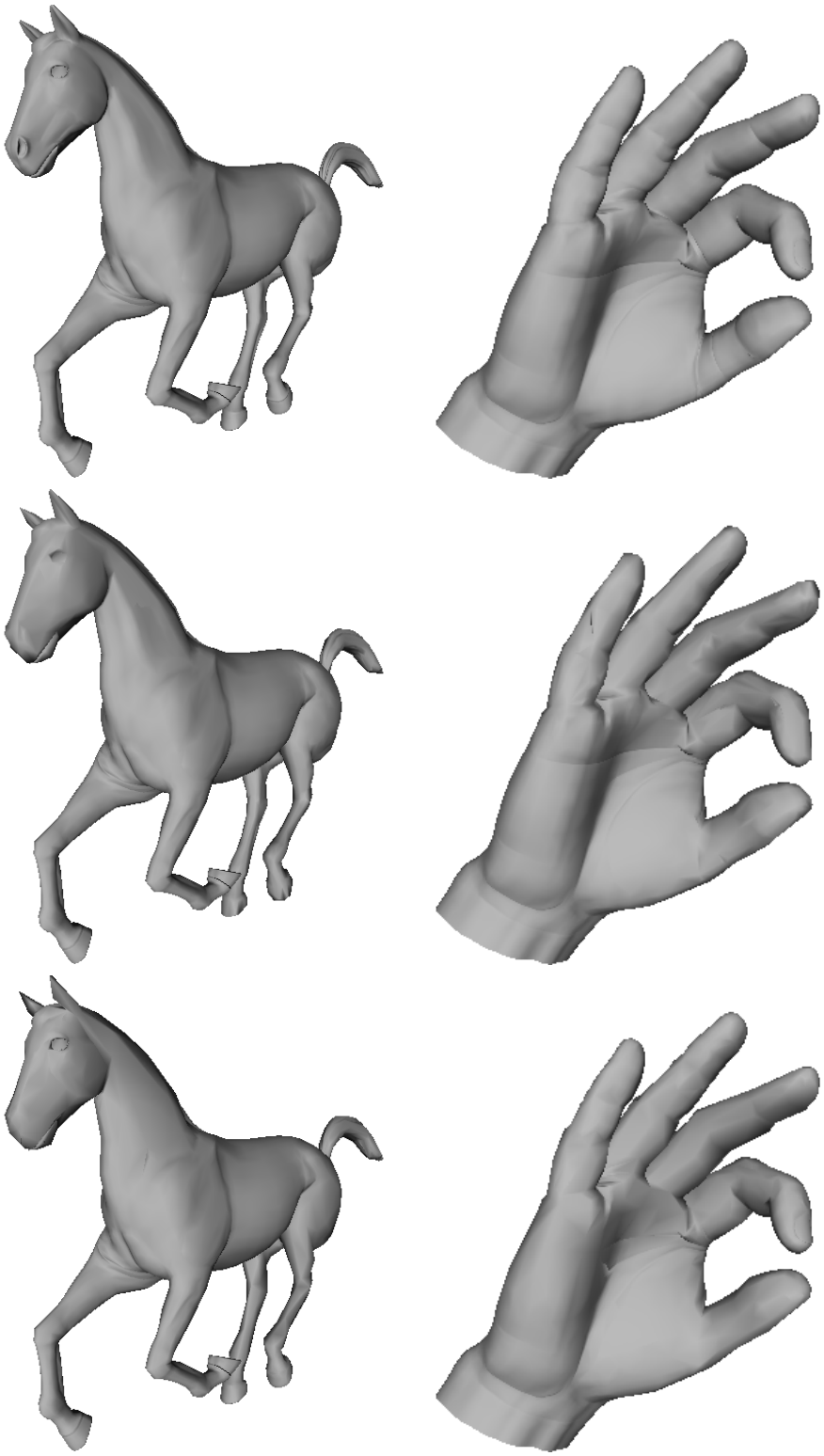


Figure 3.8: Top: reference models ; middle: simplified to half with saliency; bottom: simplified to half without saliency.

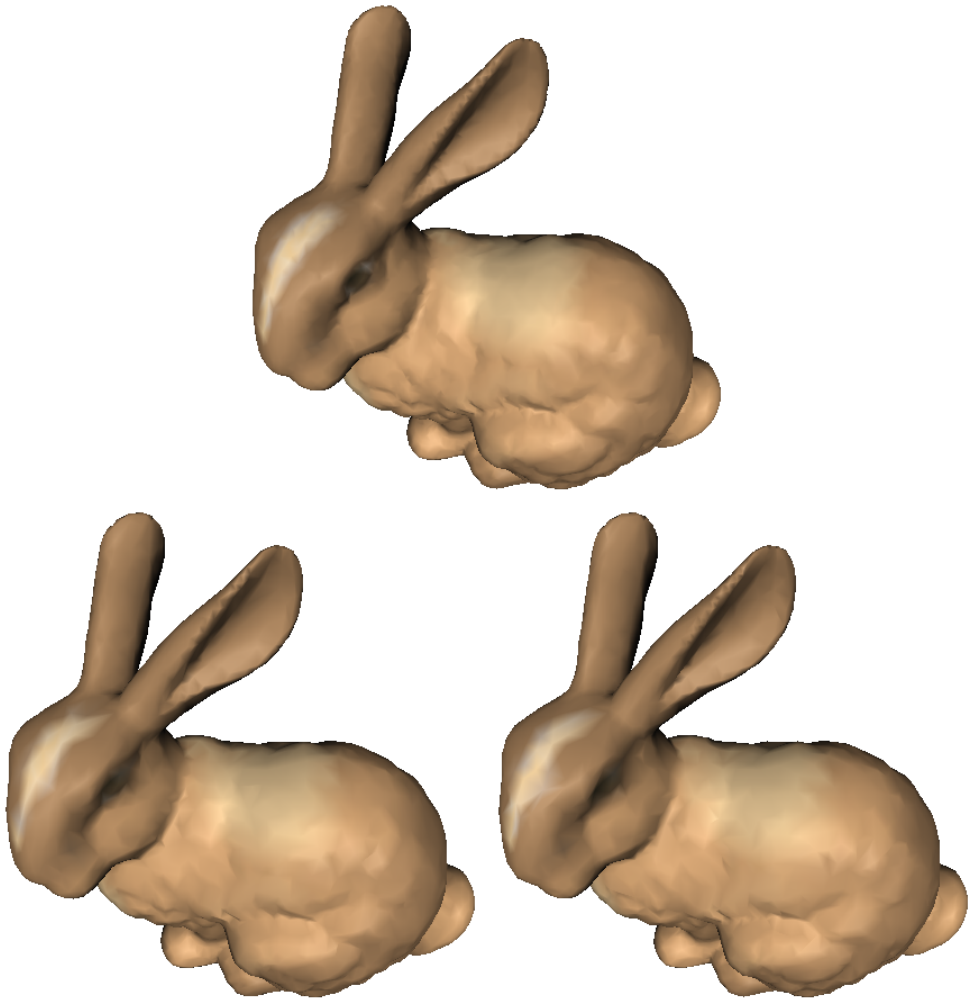


Figure 3.9: Left: reference model ; middle: simplified to half with saliency; right: simplified to half without saliency.

		# faces: 1/2 of ref.		# faces: 1/4 of ref.	
metric	**model**	**QEM**	**SGS**	**QEM**	**SGS**
HDRVDP	horse	1	0.99977	0.99966	0.99935
	hand	1	1.00003	0.99996	0.99985
PSNR	horse	1	0.97147	0.96611	0.93638
	hand	1	1.00969	0.98220	0.96735
HAUSDORFF	horse	1	0.41259	0.33333	0.15526
	hand	1	0.69791	0.48375	0.29450
MSDM2	horse	1	0.86277	0.77313	0.66278
	hand	-	-	-	-

Table 3.1: Saliency metrics and saliency guided simplification. The scores are normalized according to the score of qslim on simplifying a mesh to half number of vertices.

We have compared the performance of simplification numerically. For computing the scores of the simplified meshes we used four metrics. The details of these metrics are given in Section 2.4 except HDRVDP [84], which is an extension of VDP for high dynamic range images. Among the used metrics, PSNR and HDRVDP perform in 2D image space while Hausdorff and MSDM2 [71] work in 3D object space. As shown in Table 3.1, in most of the cases, metrics indicated that the scores are better for QEM compared to SGS. For clarity, the scores are normalized according to the score of QEM on meshes simplified to half number of vertices compared to the reference mesh.

Image based-metrics are mostly sensitive to the errors at the boundaries of the meshes due to the high contrast between foreground and background. Also, image-based metrics are affected by various factors e.g., shading methods to render the scenes, lighting conditions, shadows, and viewpoint. In this regard, viewpoint independent metrics are advantageous and result in more stable scores; nevertheless, an advantage of the viewpoint dependent metrics is that they work on the final images of the models that are actually seen by the viewers.

Despite the differences among the quality assessment metrics, the resulting scores are not much different. For the simple metrics (Hausdorff, PSNR) and for the more advanced metrics (HDRVDP and MSDM2) the ranks of the simplified meshes' scores are the same for most cases. Although we cannot generalize the results to all scenarios, we can infer that none of the used metrics could reflect the attention-based quality of meshes. Building an attention-guided quality metric could be an interesting direction of future research.

3.1.6 Discussion

We have proposed a new saliency metric to calculate the level of significance of 3D animated meshes. To be able to identify the saliencies due to different properties, the proposed metric takes various features of 3D models into account while computing the saliency. These features are related to the shape, color, and motion of an animated 3D model. The saliencies due to each feature are

computed separately, and they are normalized and combined to calculate the final saliency map. The proposed method is view independent; thus, the saliency map can be calculated for an animated 3D mesh once and can be used for both view-dependent and view-independent applications. We have also recommended several applications in which the saliency information could be useful.

3.2 Per-Object Saliency Model

In a scene with many objects, the target of our attention becomes individual objects instead of parts of objects. Extracting saliency information in such a scene could be performed on a per-object basis instead of per-vertex basis. In this section, we present our second saliency computation, Per-Object Saliency (POS) model which aims finding the saliencies of separate objects. For each animating object, a saliency value is assigned with the proposed algorithm.

By considering psychological literature given in the Sections 2.1 and 2.3, we describe a model to compare the attention values of objects in terms of motion. While motion by itself doesn't attract attention, its attributes such as initiation of a motion may attract attention. The proposed model is based on the effect of motion states on the attractiveness level of a visual stimulus.

Initially, we consider the motion attributes to discriminate different states of an object's motion. The following six states (Table 3.2) form the essence of a motion cycle (Figure 3.10):

3.2.1 Pre-experiment

We performed an eye-tracker based pre-experiment to observe the relations between the defined states and the attentively attractive objects in a 3D scene.

Static:	No change of location.
Object Appearance:	Appearance of an object, which was not previously present, on the screen.
Motion Onset:	The start of a motion (static to dynamic).
Motion Offset:	The end of a motion (dynamic to static).
Continuous Motion:	The state of keeping the motion with the same velocity.
Motion Change:	Change in the direction or speed of a motion.

Table 3.2: States of motion

In the experiment, we asked subjects to watch the animations including movements of five balls within a 3D room. Eight different animations were used (Figure 3.11 and Figure 3.12). In each of them, we analyzed different motion states and the observers' reactions to them. We used an SMI Red Eye Tracker to observe and analyze gaze points of subjects during the animations. Eight voluntary graduate students, who have reported normal or corrected to normal vision, participated in the experiments. The subjects were not told about the purpose of the experiments and each of them watched eight different animations.

3.2.1.1 Experiment cases.

The cases used in the pre-experiment were as follows:

- Case 1: Motion onset and motion offset are tested. Five initially static objects start to move in sequence with at least 3 seconds intervals. At the end, forty seconds later, all objects stop again in sequence with at least 3 seconds intervals.
- Case 2: Motion onset and motion offset are tested for shorter intervals. All objects start to move in sequence with 0.3 second intervals and stop the same way.

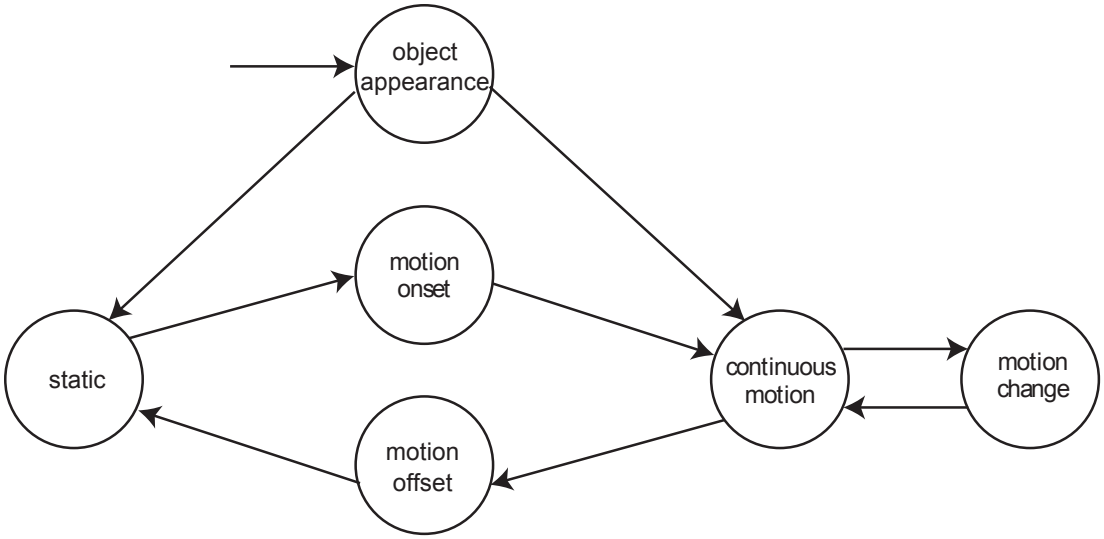


Figure 3.10: Motion cycle of an object in an animation. Perceptually distinct phases are indicated as six states. Object appearance is the initial state for each object. (From [11]. ©2011 Springer, reprinted with permission.)

- Case 3: Object appearance is tested. Some objects start to move and another object abruptly appears. There is an interval of five seconds between the latest motion onset and the object appearance.
- Case 4: Object appearance, motion onset, and motion change are tested. One object starts to move, one object appears, and another moving object changes its direction at the same time.
- Case 5: Object appearance, motion onset, and motion change are tested in larger spatial domain. The only difference from the fourth case is that the objects are placed at further points in screen space.
- Case 6: Object appearance, motion onset, and motion change are tested in sequential order. This time, three states do not occur at the same time, but in sequential order with intervals of 3 seconds.
- Case 7: Velocity difference is tested. Each object starts to move at the same time and the same direction with different velocities.
- Case 8: States are tested altogether.

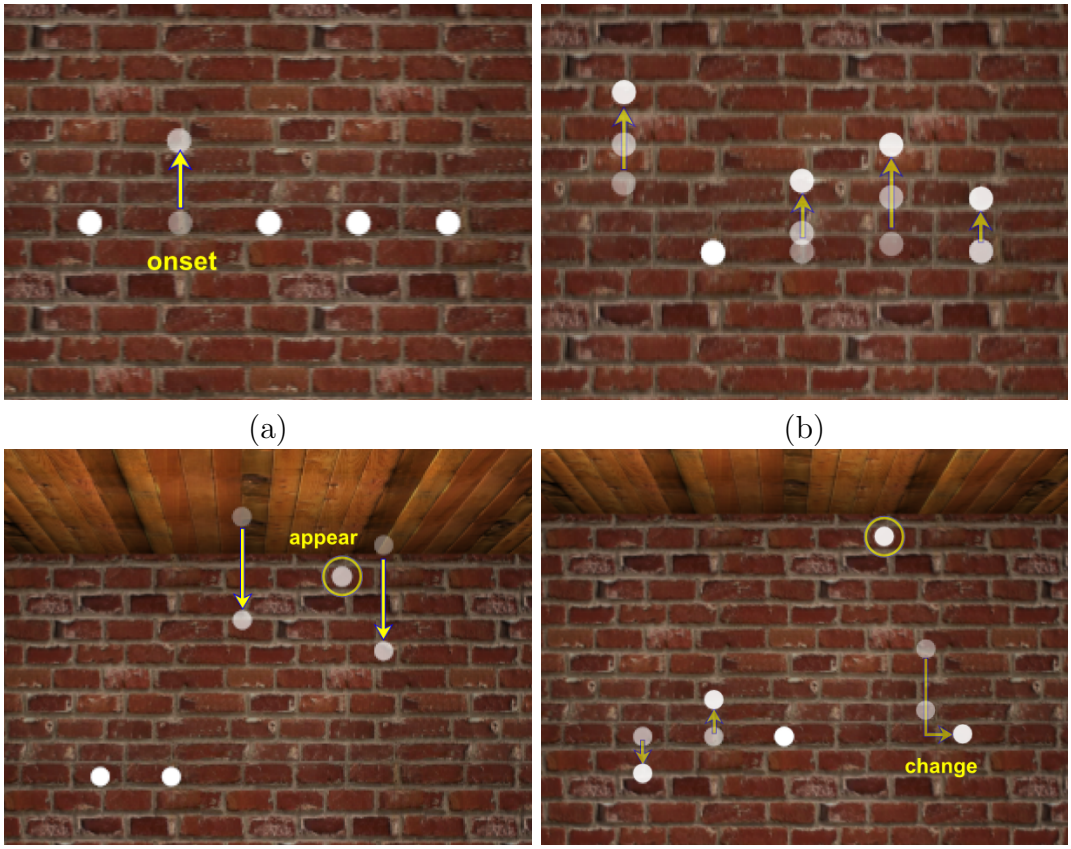


Figure 3.11: Screenshots from the eight pre-experiment animations. **a:** Motion onset and offset are tested. **b:** Motion onset and offset are tested for shorter intervals. **c:** Object appearance is tested. **d:** Object appearance, motion onset, and motion change are tested.

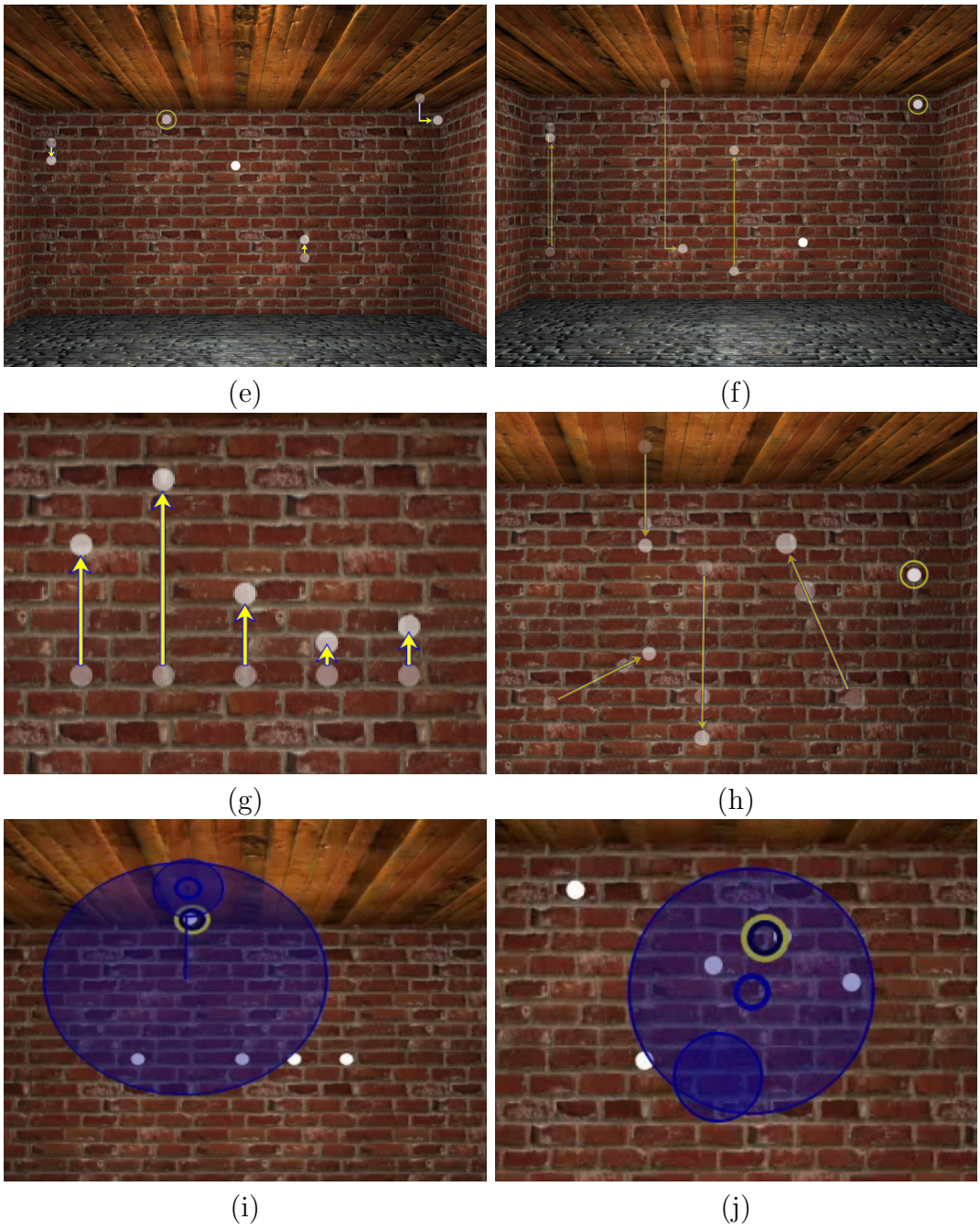


Figure 3.12: Screenshots from the eight pre-experiment animations. **e:** Object appearance, motion onset, and motion change are tested in larger spatial domain. **f:** Object appearance, motion onset, and motion change are tested in sequential order. **g:** Velocity difference is tested. **h:** States are tested altogether. **i, j:** Sample gaze positions from SMI-Red eye tracker.

3.2.1.2 Observations.

Our observations of the pre-experiment results are listed as follows:

Motion onset, *motion change*, and *object appearance* states strongly attract attention. In the first case (Figure 3.11-a), five object onsets were performed and among eight subjects, five subjects looked at all of the motion onsets. Each of the other three subjects missed only one motion onset and the missed objects were different for each of them. In the object appearance case (third animation), all subjects directed their gazes to the newly appeared object (Figure 3.11-c). In the cases four and five, none of the three objects with three different state sets dominate over the others (Figures 3.11-d, 3.12-e); indicating that *motion onset*, *motion change*, and *object appearance* states do not suppress each other. However, *motion offset*, *continuous motion*, and *static* states could be suppressed by any of the other three states.

Continuous motion and *motion offset* states capture attention merely over the *static* state. The gazes of the subjects were rarely transferred to the objects with a motion offset event. However, when we asked the subjects, they informed us that they were aware of all motion offsets but they probably did not need to move their gazes to see them. We conclude that, in most cases, subjects perceive motion offsets in their peripheral view, but their attention is not captured by these events.

In all of the experiments, none of the objects in *static* state were observed to capture attention while others were in different states. Therefore, we could say that static state does not contribute to the visual attractiveness of an object.

A further important observation is that, if the time interval between two state transitions (not for the same object) is smaller than 0.3 seconds, the transition which was set later was not captured by the subjects. In case two (Figure 3.11-b), for instance, where all objects start to move in sequential order with 0.3 seconds intervals, none of motion onsets truly captured attention of the subjects. This is

an expected behavior, since following the first event, human attentional mechanism remains on this location in the first 0.3 seconds, causing new events to be discarded [116]. After this very short period, the inhibition of return (IOR) mechanism slows the response to the current focus of attention, enabling previously unattended objects to be attended. This decay time for the state is 0.9 seconds. After 0.9 seconds, the effect of state disappears [99] (Figures 3.11-b,d; 3.12-e,f).

Gaze is transferred to the closest object upon multiple events attracting attention. In the final case (Figure 3.12-h), we observed that if more than one motion change exists, for example object appearance, motion onset, or motion change appear at the same moment, the subjects' gazes are commonly transferred to the closest object to the current gaze point.

Lastly, if more than one object have the same state, with the same speed and the same direction of motion, they are recognized as a single object according to the Gestalt principles introduced in Section 2.3. In case eight, subjects did not check each object separately in these moments; instead they looked at a point in the middle of this object group having the same motion direction and speed.

3.2.2 Overview

The overview of the proposed model is shown in Figure 3.13. The model consists of two main parts. In the first part, individual motion saliencies of the objects are calculated. In the second part, the relations among the object are examined, and the focus of attention is decided.

3.2.3 Object Motion Saliency

Based on the observations from the pre-experiments and findings from the literature in psychology related to states of motion [3, 51, 132], we propose a new individual motion saliency model. The proposed model calculates instant saliency values dynamically for each object in an animation. Once the motion state of an

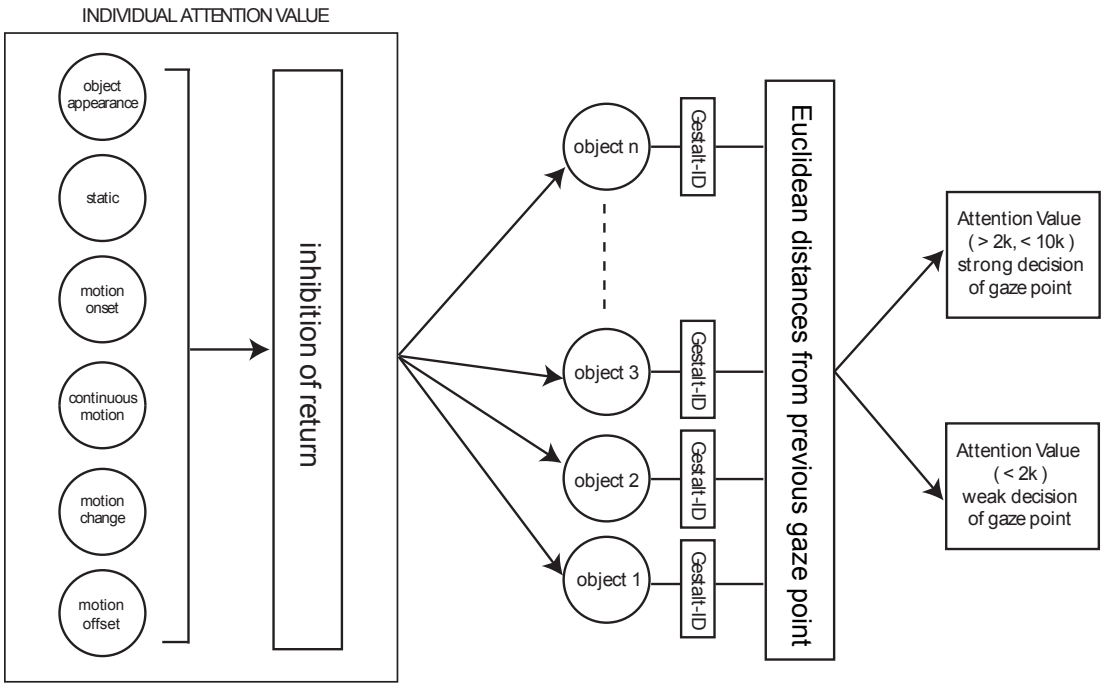


Figure 3.13: Overview of the POS model. (From [11]. ©2011 Springer, reprinted with permission.)

object is detected, its saliency value is calculated as a time dependent variable.

We define an initial saliency value for each motion state according to the dominancy among the states (Figure 3.14). These initial values are the peak saliencies, and they decay in time. Although these constants do not reflect an exact proportional dominancy result of the states among each other, they are used to estimate the saliency ranking with respect to their attentional dominancy. The initial saliency values for three most dominant states motion onset, motion change and object appearance are assigned as $10k$ while they are $2k$ for motion offset and continuous motion. For static state, it is $1k$. Usage of k as a coefficient enables converting the calculated saliencies for each object to probabilities defining the chance of getting the observers' gazes. Each visible object in a scene is a candidate for being the target of user attention and has a saliency value in terms of coefficient k. Setting the sum of all saliency values to one results in a k value, using this value saliency of an object corresponds to the calculated probability of this object to be the gaze point.

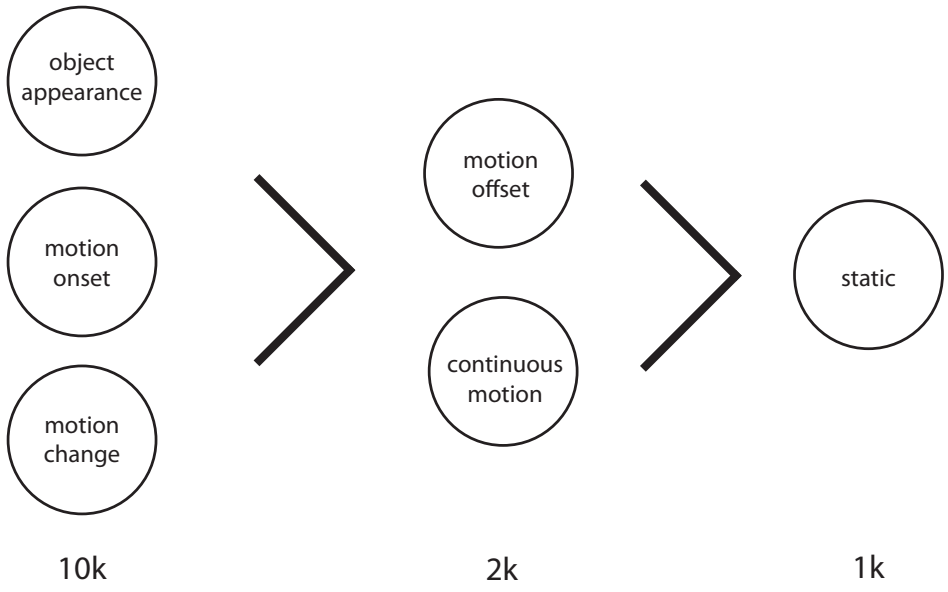


Figure 3.14: Attentional dominancy of motion states over each other. Object appearance, motion onset and motion change are the most dominant states capturing attention. Motion offset and continuous motion states are slightly more dominant over the static state. (From [11]. ©2011 Springer, reprinted with permission.)

Table 3.3: Individual Attention Values. (From [11]. ©2011 Springer, reprinted with permission.)

States	$t \leq 300ms$	$300ms < t < 900ms$	$900ms \leq t$
STATIC	$1k$	$1k$	$1k$
ONSET	$10k$	$10k((4.2 - 4t)/3sec)$	$2k$
OFFSET	$2k$	$2k((7.5 - 5t)/6sec)$	$1k$
CONTINUOUS	$2k$	$2k$	$2k$
CHANGE	$10k$	$10k((4.2 - 4t)/3sec)$	2k
APPEARANCE	$10k$	$10k((2.9 - 3t)/2sec)$	$1k$

For each visible object, calculated individual saliency values change between $1k$ and $10k$ according to the formulas shown in Table 3.3, where t stands for the elapsed time after a state is initialized. Obviously, the saliency value for invisible objects is zero.

Saliency of an object in the static state is always $1k$, because we observed that it almost never captures attention among other states. Also, saliency value is permanently $2k$ for continuous motion since a moving object may get attention anytime if nothing interesting, e.g. states of all other objects are static, happens on the screen and the subject is not performing a target search [132]. Likewise, motion offset could get attention over static or continuous motion. However, it is under decay with the effect of IOR until the state becomes static.

Motion onset changes to continuous motion state; therefore, its value decays to $2k$ with IOR. It is exactly the same for motion change state. Hence, during inhibition of return (0.3sec$<$ t $<$ 0.9sec) [5] [6], the attention value decays linearly from $10k$ to $2k$ as shown on Table 3.3. Differently, object appearance decays to $1k$ because its state may become static. Similarly, motion offset decays linearly from its peak value $2k$ to $1k$, since its following state is static.

Inhibition of return is not applied to any of the states if the elapsed time on a state is smaller than 300ms. This is because the sensitivity to an attended region is kept at the top-level for the first 300ms. Before IOR decreases our sensitivity

to this location, our attention remains here. As we mentioned earlier, during this time, other state changes are not captured by the subjects in the experiments.

This model provides calculating individual motion saliency values of each object in real time. However, these values are still not sufficient to decide the focus of attention without examining the relationships of the objects among each other.

3.2.4 Global Attention Value

So far we have shown a model to calculate individual attention values of the objects. On the other hand, our visual system does not interpret objects in the scene individually. It tries to group the similar behaviors in the visual scene and represent any stimulus in the simplest way. As Gestalt psychologists concluded, all similar objects in our vision are grouped and perceived as a simple object [9]. We, therefore, include a Gestalt organization layer into our model. Each object in this model has a *Gestalt ID*. In this work, we assume that, the objects having identical motion direction and speed are labeled with the same Gestalt ID. The pseudocode for this procedure is as follows.

Algorithm 1 Setting Gestalt ID's to objects

```
curGestalt = 1
for i = 1 to NUMBEROFOBJECTS do
  gestaltSet = false
  for j = 1 to i-1 do
    if objects[j].velocity = objects[i].velocity then
      objects[i].gestaltID = objects[j].gestaltID
      gestaltSet = true
      break
    end if
  end for
  if !gestaltSet then
    objects[i].gestaltID = curGestalt
    curGestalt++
  end if
end for
```

In a group of objects having the same Gestalt ID, the object with the highest saliency determines the saliencies of all objects in this group. In the pre-experiments, we clearly observed that if there are several objects with the same speed and direction, subjects' gazes circulate among the area covered by the group members, instead of looking directly at individual objects revealing the perceptual unification of these objects.

Another problem that is not solved by the individual saliency model is the case of equivalence. If there are multiple objects with the highest individual attention value, which object will be chosen as the possible gaze point is not addressed. An observation we made during pre-experiments suggest a solution to this problem. If more than one state changes occur at the same time, subjects commonly looked at the closest object to the previous gaze point. Therefore, we included an attribute to our model to consider the Euclidean distance of each object from the previous decision of gaze point. It is calculated as the pixelwise Euclidean distance from the previously decided focus of attention in the screen.

Finally, if there are multiple Gestalt groups with the highest calculated saliency, the closest one to the previously decided focus of attention is selected as the current focus of attention.

3.3 Extended Per-Vertex Saliency Model

The POS model proposed in Section 3.2 is applicable to separate objects in a 3D animation scene. In this model, a saliency value is calculated for each object. In this section, we propose a technique to extend the proposed per-object saliency calculation model to 3D meshes, where the saliency property is calculated in a per-vertex fashion.

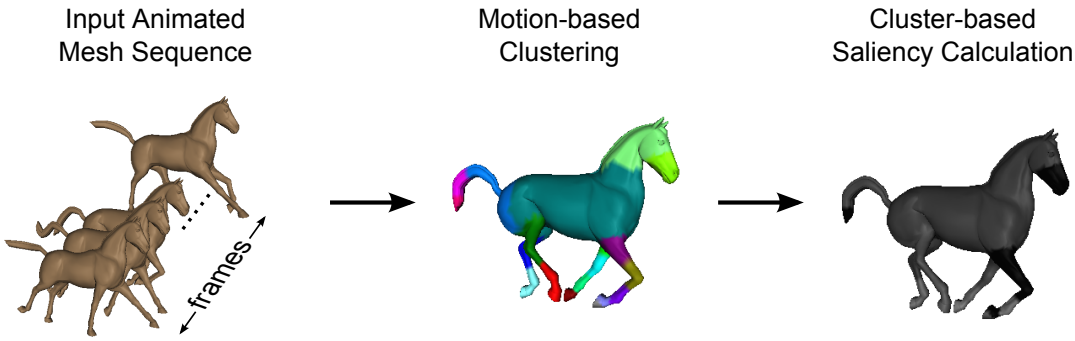


Figure 3.15: Overview of the cluster-based saliency calculation model.

3.3.1 Overview

The simplest approach to extend the per-object saliency calculation model to a per-vertex saliency calculation model for 3D meshes is regarding each vertex as an object. However, this approach would be computationally expensive. Furthermore, looking at a 3D mesh, we do not recognize each vertex as a separate part of an object; we rather group the similar vertices perceptually and regard each groups as a single unit. This enables common analysis of vertices that move together.

Our proposed approach is based on clustering the vertices with a similar behaviour. Figure 3.15 illustrates an overview of the model. In our model, each frame of a vertex-animated mesh sequence is analyzed to cluster vertices according to their motion properties. As the vertices with similar motion behaviour are perceptually grouped with a Gestalt approach, each cluster is analyzed as a single unit after the clustering process. Thus, representative vertices for each cluster are used to extract the motion saliencies of their clusters. Since the number of vertices to be analyzed reduces significantly (from total number of vertices to number of clusters) via the clustering process, the saliency calculation can be performed in real time.

3.3.2 Motion-based Clustering

In this clustering technique, our aim is to group the vertices with similar motion into the same cluster. Thus, the generated clusters should contain vertices that have very close motions through the animation. Additionally, we need to form clusters in which all vertices are connected. For this purpose, we utilize the velocities of vertices and their connectivity information. The clustering procedure is summarized in Algorithm 2:

Algorithm 2 Pseudocode for motion-based clustering

```
for all frames of the animation do
    Find differential velocities of all vertices
    Extract boundary vertices using differential velocities
    Assign vertices inside a bounded area to a separate cluster
end for
Refine clusters to get final clustering
```

3.3.2.1 Finding Differential Velocities

In this step, our aim is to find the vertices that have different velocities compared to their surroundings. Since vertices in a group have similar velocities through the animation, we assume that the vertices having high relative velocities to their neighbors resides in the boundaries between clusters. We name the difference between the velocity of a vertex v and the Gaussian-weighted mean velocity in its surrounding region the *differential velocity* of vertex v ($dv(v)$), calculated as fallows:

$$\vec{dv}(v) = |\vec{vel}(v) - G(\vec{vel}, s, v)|, \tag{3.9}$$

where, $\vec{vel}(v)$ is the velocity of v calculated as in Equation 3.1 and $G(\vec{vel}, s, v)$ is the Gaussian weighted mean velocity of the surround s of v. For s, we used 0.036 of the size of the mesh's bounding box as the radius of surrounding region. The differential velocity calculations are done on a vector basis instead of considering scalar values, to be able to differentiate velocities in different directions.

Figure 3.16: Differential velocities on a 3D model. Brighter (yellow) regions express high differential velocities. The figure shows the absolute amounts of differential velocities in a scalar manner for a better presentation.

Figure 3.16 illustrates differential velocities on a 3D model. As shown in this figure, differential velocities are higher in the boundary regions that separate the concrete regions that move together.

3.3.2.2 Extracting Boundary Vertices

After calculating the differential velocities of vertices, those vertices that have high differential velocities are selected as the boundary vertices. The set of boundary vertices BV is composed of the vertices having higher differential velocities than a threshold.

$$BV = v \in V || dv(v)| > t_{bound}, \tag{3.10}$$

where V is the set of all vertices in the mesh and t_{bound} is the lower threshold for the absolute differential velocities of boundary vertices. The vertices in BV will be used to form clusters; thus, having more boundaries result in more clusters. Selecting a high t_{bound} value results in less boundary vertices and less number of clusters. In the opposite case, there would be more boundary vertices and more

clusters. Having less number of clusters is worse than having redundant clusters since less clusters result in wrong saliency calculations while redundant clusters is safe but result in computational overhead. Therefore, we selected a low value for t_{bound} as 0.001 percent of the maximum absolute differential velocity among all vertices of the mesh.

3.3.2.3 Forming Temporary Clusters

In each frame of the animation, boundary vertices are used to form clusters, and the generated clusters are merged with the clusters from the previous frame. This way, clusters are accumulated through the animation and every single motion in any frame forms a cluster in the final clustering. The pseudocode to form the clusters in a single frame is shown in Algorithm 3.

Algorithm 3 Generating temporary clusters at each frame

```
initialize NonClusteredVertices with all vertices except boundaries
initialize currentCluster with 1
while NonClusteredVertices is not empty do
  initialize TraceList
  vertex v = first element in NonClusteredVertices
  assign v to currentCluster
  remove v from NonClusteredVertices
  add v to TraceList
  while TraceList is not Empty do
    for all neighbors n of v do
      if n is not boundary then
        assign n to currentCluster
        add n to TraceList
        remove n from NonClusteredVertices
      end if
    end for
    remove v from TraceList
    v = first element of TraceList
  end while
  increment currentCluster
end while
```

Figure 3.17 shows samples from the temporary clusters through an animation.

Figure 3.17: Clustering through an animation, from left to right (except the rightmost): clustering results after several frames are shown. White regions depict the boundary vertices. The rightmost image shows the final clustering after clustering refinement phase.

As shown in this figure, the boundary vertices are not assigned to any cluster in this phase, they will be assigned to the closest clusters to them in the cluster refinement phase.

3.3.2.4 Cluster Refinement

After temporary clusters are set for all frames, clusters are refined to form the final clustering. In this phase of the method, each vertex is assigned to the cluster with the most similar motion behavior to this vertex through the animation. The distance of a vertex v to a cluster C is calculated by summing up the squared differences between the velocity of v and mean velocity of vertices belonging to C at each frame. This distance calculation is shown in the following equation.

$$dist(v, C) = \sum_{f=1}^{F} \left| \vec{vel}(v, f) - \frac{\sum_{w \in C} \vec{vel}(w, f)}{|C|} \right|^2, \tag{3.11}$$

where F is the number of frames through the animation and $\vec{vel}(v, f)$ is the velocity of vertex v in frame f. The following algorithm is performed to finalize the clustering process. Several clustering results are presented in Figure 3.18

Algorithm 4 Generating temporary clusters on each frame

while There is an update in last iteration **do**
  for all $e \in Edges$ **do**
    $v1$ and $v2$ are the vertices connected by e
    if $v1.cluster \neq v2.cluster$ **then**
      $d_{V1toC1} = dist(v1, v1.cluster)$
      $d_{V1toC2} = dist(v1, v2.cluster)$
      $d_{V2toC1} = dist(v2, v1.cluster)$
      $d_{V2toC2} = dist(v2, v2.cluster)$
      if $d_{V1toC1} > d_{V1toC2}$ **then**
        assign $v1$ to $v2.cluster$
      end if
      if $d_{V2toC1} < d_{V2toC2}$ **then**
        assign $v2$ to $v1.cluster$
      end if
    end if
  end for
end while

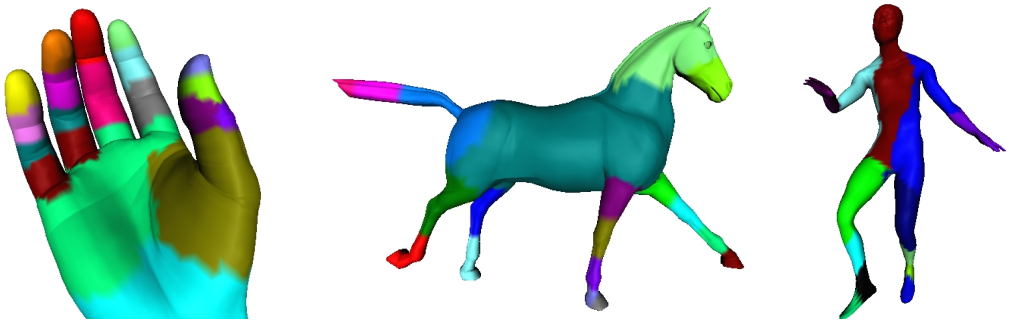


Figure 3.18: Clustering results for several 3D models.

3.3.3 Saliency Calculation for Clusters

3.3.3.1 Identifying Cluster Heads

In the EPVS model, the approach proposed for the POS model proposed in Section 3.2 is utilized to calculate the saliency values for each cluster. In this process, each cluster is handled as an object. For each cluster, firstly, we select a representative vertex, cluster head. Then, the motions of the cluster heads are analyzed through the animation to calculate the saliency of their clusters.

A cluster head is a selected vertex with the smallest distance to the cluster centroid according to the distance formula given in Equation 3.11. By cluster centroid, we mean an imaginary vertex having the average velocity of all vertices in a cluster at each frame of the animation. Cluster heads represent their clusters and since number of cluster heads is much less than the number of vertices, calculating saliency on them could be done in real time.

3.3.3.2 Calculating Saliency

The motion state of a cluster head is extracted using its velocity relative to the whole object. The motion that is caused by the movement of the entire object should be avoided in saliency calculations for clusters; thus, we subtract the average velocity of all clusters from the velocities of the cluster heads to obtain their relative velocities as follows:

$$rv(\vec{v}, f) = vel(\vec{v}, f) - \frac{\sum_{w \in V} vel(\vec{w}, f)}{|V|}, \tag{3.12}$$

where V is the set of all vertices and f is a frame of the animation sequence.

Relative velocities of the cluster heads and two threshold values, t_{onset} and t_{change}, are used to extract the motion states. The motion states are calculated with the following conditions:

$$State(v) = \begin{cases} |rv(v,f)| > t_{onset} \& |rv(v,f-1)| < t_{onset} & \Rightarrow & MotionOnset \\ |rv(v,f)| < t_{onset} \& |rv(v,f-1)| > t_{onset} & \Rightarrow & MotionOffset \\ |rv(v,f) - rv(v,f-1)| > t_{change} & \Rightarrow & MotionChange \\ |rv(v,f)| > t_{onset} & \Rightarrow & Continuous \\ otherwise & & Static \end{cases} \tag{3.13}$$

Human visual system is not good at recognizing speed change [93], this is the reason behind having a different threshold for motion change in our algorithm. In general, we set t_{change} as $2 * t_{onset}$ and t_{onset} as follows:

$$t_{onset} = \tan 0.15^{\circ} * d_{user} * k_{wtoc}, \tag{3.14}$$

where 0.15° is the drift velocity of the eye that could be traced like a static situation [134] and k_{wtoc} is a constant to convert the result from world space to computer space.

Each state is related to a saliency value, which are identical to the values shown in Table 3.3. Saliency values are assigned to vertices according to this table based on their motion states. Furthermore, according to our observations, a strong motion onset (starting with a higher velocity), attracts more attention so we have weighted the saliencies according to the relative velocities of the cluster heads when a motion onset or motion change occurs. Equation 3.15 shows this operation:

$$saliency(v)' = \frac{(saliency(v) - s_{base}) * |rv(v)|}{rv_{max}}, \tag{3.15}$$

where rv_{max} is the highest relative velocity at the moment and s_{base} is the saliency value given to continuously moving vertices, shows this operation:

When a motion onset or motion offset occurs, the attention to this region remains for approximately 0.3s and disappears in 0.9s according to the inhibition of return mechanism as explained in Section 2.1.1.3. Thus, when a state change is

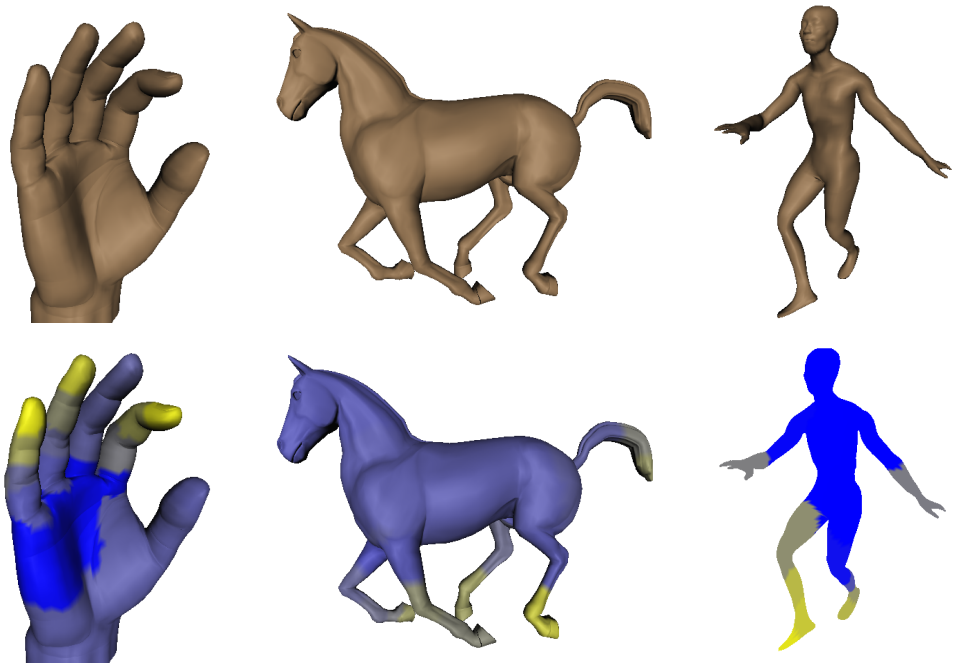


Figure 3.19: Calculated saliencies on 3D models. Bottom: brighter (yellow) regions show more salient parts of the models on the top.

identified, its effect is calculated for the following frames according to the formulas in Table 3.3. When a new state change occurs before the effect of previous motion state vanishes, a cluster head is affected by two motion states at the same time. In such cases, the maximum saliency value is selected as the final saliency value.

After calculating the saliency values for cluster heads, their saliency values are spread to all vertices in their clusters. Figure 3.19 shows calculated saliencies on several 3D models.

For saliency calculations, we use velocities in 3D space instead of using the 2D velocities projected to the screen space as the human visual system could extract the absolute 3D velocity from the 2D retinal images [38]. However, since the proposed saliency calculation model could work in real-time, retinal velocities of the cluster heads could also be calculated and used for saliency calculations.

Chapter 4

Attention-based Stereoscopic Rendering Optimization

4.1 Mixed Stereoscopic Rendering

In this chapter, we present a binocular suppression theory based approach to stereoscopic graphics rendering. The proposed method exploits the fact that the 3D perception of the overall stereo pair in a region is determined by the dominant image on the corresponding region, instead of summation of the effect of two images. The dominant view varies in different parts of the image, forming a mosaic pattern of suppression, creating the overall 3D percept. Our goal is to explore how the rendering quality can be reduced in the suppressed view, without reducing the overall perceived quality of the rendered 3D image. If such features can be detected, rendering computations of those features for one eye can be reduced, thus increasing the overall speed of rendering.

Figure 4.1 illustrates the traditional and the newly proposed mixed stereoscopic rendering approaches. In the traditional approach, the left and right views are generated with the same rendering technique and the same quality. On the other hand, in the proposed approach, two views can have different parameters of rendering - one of the views is generated with the original quality and the other

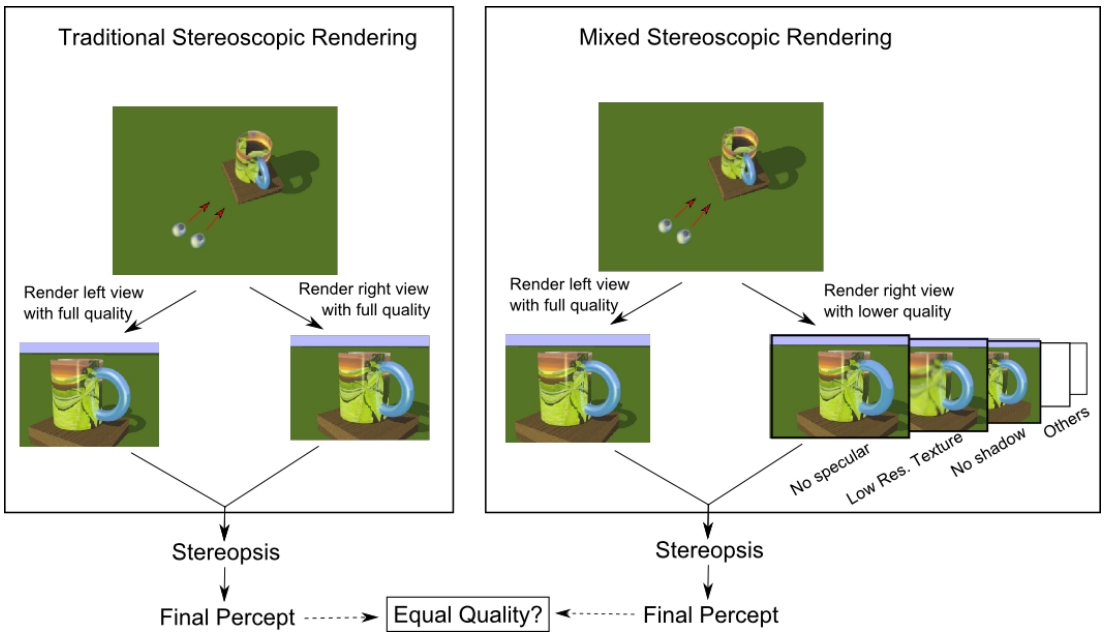


Figure 4.1: Left: Traditional stereoscopic rendering approach, Right: Our rendering approach for optimization. On the upper images, the object space is illustrated along with the viewpoint. The images below show the corresponding left and right eye views of the above scene. The right view on the mixed stereoscopic rendering approach is generated with lower quality caused by simplifying a set of modeling and rendering features. (From [20]. ©2010 Elsevier, reprinted with permission.)

view with lower quality, thus decreasing the overall rendering cost. However, this approach is feasible when the overall quality of the final 3D percept is determined by the high quality image.

4.1.1 Mixed Stereo Methods

The proposed mixed stereoscopic rendering method manipulates various graphics rendering and modeling conditions, in different levels of intensity. In this work, a number of representative and commonly used image and geometry based methods, which are employed in virtual environments, have been investigated. These methods include *framebuffer upsampling*, *blurring*, *mixed-level antialiasing*, *texture resampling*, *mesh simplification*, *mixed shadowing*, *specular highlight*, and *mixed shading*. Table 4.1 and the rest of this section illustrate the used methods

in detail. The applied methods are classified into two classes: 2D image-based methods, and 3D object/rendering-based methods.

2D image-based methods

2D image-based methods - including framebuffer upsampling, blurring - are methods that operate on the framebuffer. The main purpose of the 2D image-based methods is to decrease the rasterization cost of one view while keeping the overall quality. In these methods, an image for one view is generated with low quality, and then image-based post processing operations are performed on it.

Framebuffer Upsampling: In framebuffer upsampling, the 3D scene is rendered to a smaller framebuffer in one view, and then this buffer is upsampled to match the framebuffer resolution for the high-quality view. Four different sizes of framebuffers have been used, each level halving the width and height of the previous level (1/4 area of the previous size). For upsampling, the Lanczos resampling algorithm has been used [118].

Blurring: As demonstrated in Section 2.1, a widely-used 2D technique is applying a rendering effect for approximate rendering. In this work, a blur filter is used to demonstrate its effect. For blurring the images, a Gaussian filter is used, with radiuses of different sizes which determine the strength of the filter [36]. The radiuses of Gaussian filters used for levels 2, 3 and 4 are 1, 3, 5 pixels respectively, where the first level is the original framebuffer (Table 4.1). The σ values of the Gaussian filters are the same for each level, and are chosen as 1.

3D object/rendering based methods

3D object/rendering-based methods operate on the 3D scene, i.e. application stage of the rendering pipeline (mesh simplification, mixed shadowing), or on subsequent stages of the rendering pipeline (mixed-level antialiasing, texture resampling, specular highlight, mixed shading). The general goal of these methods is to decrease the rendering pipeline computations on the CPU and the GPU, by excluding a number of rendering effects for only one of the views.

Mixed-level antialiasing: Antialiasing, based on sampling more than one

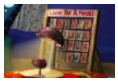
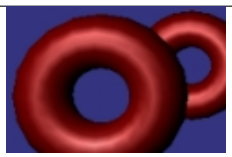
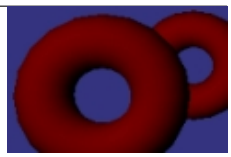
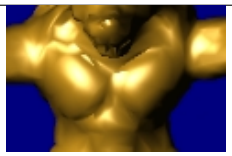
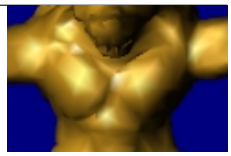
Method	Levels	A	B
Framebuffer upsampling	Lvl 1: Original (A) Lvl 2: 1/2 Size (B) Lvl 3: 1/4 Size Lvl 4: 1/8 Size		
Blurring	Lvl 1: Original (A) Lvl 2: Blurred (filter radius 1) (B) Lvl 3: Blurred (filter radius 3) Lvl 4: Blurred (filter radius 5)		
Mixed-Lvl antialiasing	On: Antialised (A) Off: Not-antialiased (B)		
Specular highlight	On: with spec. highlight (A) Off: without spec. highlight (B)		
Mixed shading	Lvl 1: Phong shaded (A) Lvl 2: Gouraud shaded (B)		
Mesh simplification	Lvl 1: Original Mesh (A) Lvl 2: Face count = 1/2 of orig. Lvl 3: Face count = 1/4 of orig. (B) Lvl 4: Face count = 1/8 of orig.		
Texture resampling	Lvl 1: Texture size 512 x 512 (A) Lvl 2: Texture size 256 x 256 Lvl 3: Texture size 128 x 128 Lvl 4: Texture size 64 x 64 (B)		
Mixed shadowing	On: Shadows are used (A) Off: Shadows are not used (B)		

Table 4.1: Methods used for Mixed Stereoscopic Rendering (From [20]. ©2010 Elsevier, reprinted with permission.)

sample per pixel, is widely used in graphics applications. Different sampling patterns have been proposed, including Grid, Checker, Quincunx sampling schemes [6]. Although hardware-based antialiasing solutions are fast, processing of more than one sample per pixel is still required. Therefore, in the mixed antialiasing method, one of the views is rendered using antialiasing, and the other view is rendered with antialiasing turned off. Although different antialiasing schemes could also be used for the two views, one of the views is rendered with no antialiasing, to better illustrate its effect. In this work, a 3×3 grid super-sampling is used as the antialiasing method [6].

Specular highlight: Specular highlight is the bright spot on a reflector surface caused by the reflection of light, which depends on the viewing angle. In computer graphics, specular highlight is simulated in various specular reflectance models such as Phong, Cook-Torrance, etc. [92]. In this work, the Phong model is used to exhibit the specular highlight for one view, and the specular component of the material for the other view is ignored, resulting in a pure Lambertian reflectance model [46].

Mixed Shading: In order to investigate the effects of illuminating two views with different interpolation methods, we implemented Phong and Gouraud shading which are widely used in computer graphics. In Phong shading, normals are interpolated when calculating the color values inside a polygon in order to obtain a smooth appearance [98]; whereas in Gouraud shading only the colors are interpolated with lower cost [41]. Although a wide variety of advanced shading techniques, such as the use of BRDF, have been recently used, we chose two widely-used solutions for illustrating the effect of mixed shading.

Mesh simplification: In order to find out the effects of object-based techniques, mesh simplification is employed for all objects in one view. The simplification is done using the Quadric Edge Collapse method described in [40]. The number of faces of a mesh in a level is approximately half of the number of faces in the previous level (Table 4.1).

Texture resampling: To verify the effect of mixed-resolution textures in a scene, a texture resampling method is employed. In this method, the textures in

the scene are rendered with lower resolution in one view. Various levels of texture resampling have been used: the size of the texture map used for a level is half of the previous level in terms of both width and height (Table 4.1). Linear filtering is used for resampling of the texture images [47]. In this work, further methods for antialiasing of textures, such as mipmapping or anisotropic filtering, are not tested, to verify only the resampling effect.

Mixed Shadowing: As adding shadows to a 3D scene requires expensive calculations, avoiding these calculations for one view without affecting the final 3D percept would be a desirable optimization. In the Mixed Shadowing method, no shadow is used for one view; and point light sources with hard shadows are used for the second view [5].

4.2 Saliency-guided Stereoscopic Rendering Optimization

4.2.1 Intensity Contrast

According to the binocular suppression theory, the methods described above are effective when the regions in the simplified view are suppressed by the high-quality view. Therefore, the properties of an image, which allow a view to be suppressed, and which therefore keep the modification unnoticed, should be characterized.

Previous studies suggest that stronger competitors (e.g: higher spatial frequency, more color variance or a faster motion) are more likely to suppress [17], [56]. According to our observations, the properties that make the competitors of the binocular rivalry stronger are also the features that attract visual attention: The regions which attract more attention are likely to be the candidates for being strong competitors in a stereo pair. Itti and Koch state that the visual attention is selective, and eye gaze is oriented towards regions that show large contrast, and these regions can be defined as salient [61]. According to Itti, a region is more salient - thus attracts more attention - when it differs from its surroundings

regarding a number of properties, such as intensity, color opponency, motion, and orientation [61]. These properties are consistent with the properties that increase the strength of the competitor.

To measure the strength of a view, we use a heuristic, intensity contrast, and obtain the change in the image intensity contrast caused by a modification, to decide whether it is sufficient to apply the modification to only a single view. For this purpose, we have followed the saliency calculation method described in [60], in which a center-surround mechanism is used to compute three separate saliency maps for three channels: intensity, color and orientation. In our work, only intensity maps are needed, since the methods used for modification do not have a significant effect on the color and orientation attributes of the stereoscopic image.

4.2.2 Calculating Intensity Contrast

The first step in calculating the intensity contrast map of an image is extracting the intensity map of the image. The average of the RGB values in a pixel gives the intensity value:

$$I = \frac{R + G + B}{3} \tag{4.1}$$

The intensity contrast map is generated from the intensity image using the center surround operator described in [60]. In this method, DoG (Difference of Gaussian) filters are calculated as the difference of Gaussian filters in fine (center) and coarse (surround) scales. The center consists of the pixels with a closer distance than c and the surround consists of the pixels with a closer distance than $s = c + \delta$, where $c \in 2, 3, 4$ and $\delta \in 3, 4$. Thus, six DoG filters are calculated using fine and coarse scales as $\{2-5, 2-6, 3-6, 3-7, 4-7, 4-8\}$ (Figure 4.2) and each of them is used to generate an intensity contrast map. These six maps are added pixel by pixel to construct a final intensity contrast map.

Figure 4.3 shows a pair of intensity contrast maps and their difference. In

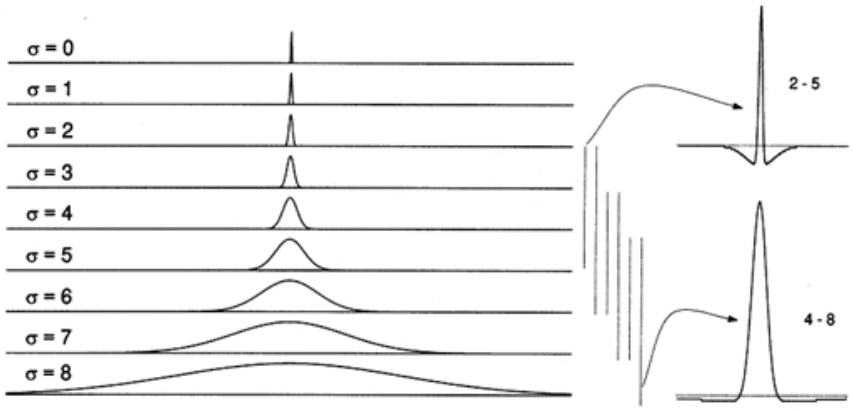


Figure 4.2: Gaussian pixel widths for the nine scales used in the intensity contrast calculation. Scale $\sigma = 0$ corresponds to the original image, and each subsequent scale is coarser by a factor 2. On the right, two examples of the six center-surround DoG filters are shown, for scale pairs 2-5 and 4-8. (From [60]. ©1998 Elsevier, reprinted with permission.)

this figure, brighter regions in the intensity contrast maps show the parts with greater intensity contrast. To calculate the effect of a modification on the intensity contrast, the intensity contrast map of the modified image (rendered with low quality) is subtracted from the intensity contrast map of the original image. In Figure 4.3, the positive values are colored as blue and the negative values are colored as yellow. The negative values on the result (yellow regions in the figure) indicate an increase in the intensity contrast due to the modification.

4.2.3 Mixed Rendering Approach

We are proposing that if the intensity contrast of a view is greater than the other view, it has the privilege of being dominant. Hence, we are hypothesizing that a modification which raises the intensity contrast needs not be applied on both of the views. On the other hand, if the intensity contrast of the modified view is lower than the original image, then the optimized pair provides the same percept as the result of traditional rendering. Figure 4.4 summarizes our hypothesis.

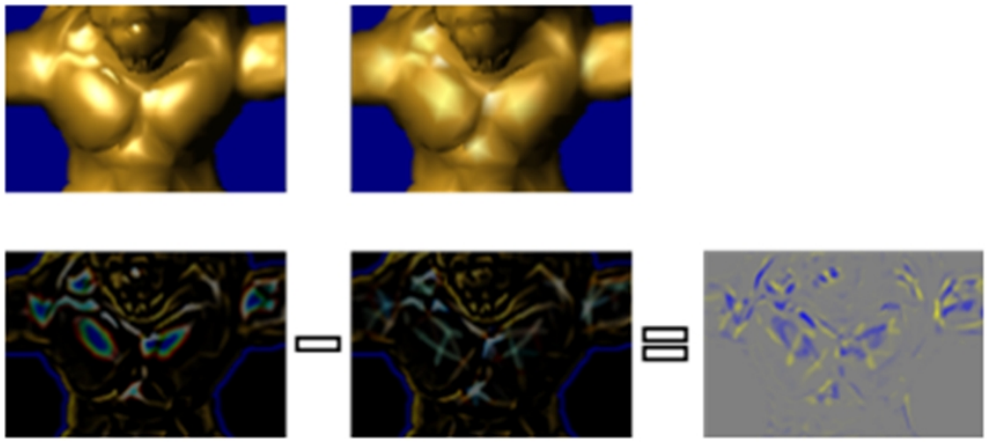


Figure 4.3: Top Left: Original image, Top Right: Modified image, Bottom left: Intensity contrast map of the original image, Bottom Middle: Intensity contrast map of the modified image (Brighter regions are the parts with higher intensity contrast in the intensity contrast maps.) Bottom Right: Calculation of Intensity Contrast Change. Difference of the two intensity contrast maps results in the right-most image, where the blue regions are the parts that the intensity contrast is greater in the original image and yellow regions are the opposite. (From [20]. ©2010 Elsevier, reprinted with permission.)

Figure 4.5 contains sample intensity contrast change maps of the applied methods. According to our hypothesis, a modification is not recognizable in blue regions of the intensity contrast change map. For instance, the intensity contrast change map of the specular highlight method contains only blue; therefore we expect that the image with the specular highlight will suppress the other image in the final percept, and this method is suitable for our optimization approach. In framebuffer upsampling, blurring and texture resampling methods; blue regions are considerably more than the yellow regions which also lead us to expect that original images are dominant in general. Thus, these methods are expected to be appropriate for stereoscopic rendering optimization. The yellow regions cover a large area in the figures of antialiasing and mesh simplification methods, therefore these methods are not suitable for our optimization approach, according to our hypothesis. For mixed shadowing, although the shadowed image has apparently more intensity contrast in the borders of the shadow; the opposite holds for the interior parts of the shadow. Hence, shadow is expected to be a strong factor for suppression, therefore it should be applied on both of the views. For mixed

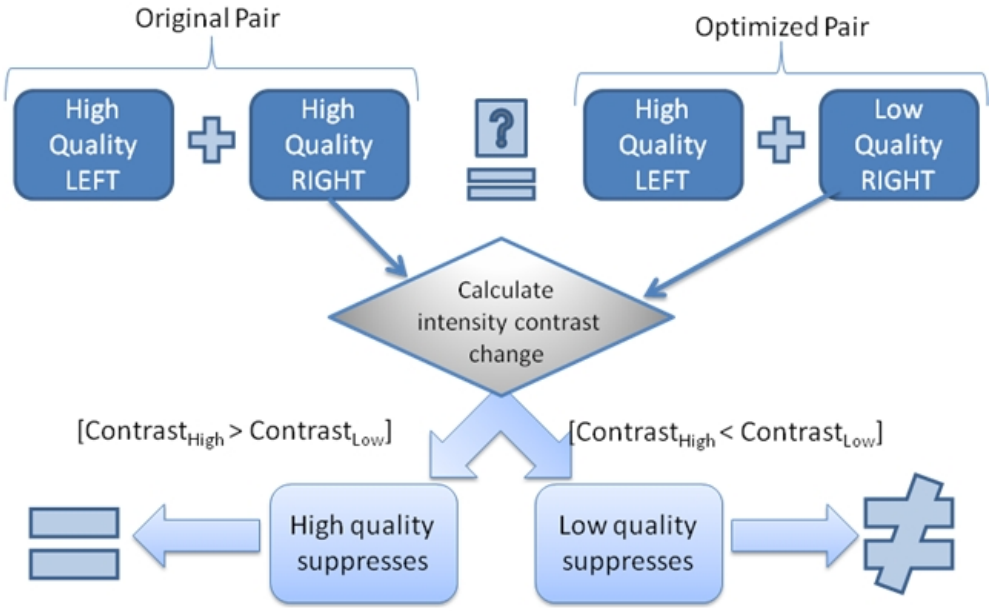


Figure 4.4: Summary of the hypothesis. Optimized pair provides the same percept if the intensity contrast of the modified view is lower than the original (ContrastHigh: Intensity contrast of high quality image). Note that, the low quality image does not have to be the right view always. It is also possible to decrease the quality of the left view. (From [20]. ©2010 Elsevier, reprinted with permission.)

shading, even though there are regions in which the Gouraud shaded image is dominant, Phong shaded image is stronger in general.

4.3 Summary of the Proposed Method

In this chapter, we have presented a perceptually-based optimization approach for stereoscopic rendering, which makes use of the binocular suppression mechanism of the human visual system. The proposed method exploits the fact that the 3D perception of the overall stereo pair in a region is determined by the dominant image on the corresponding region, instead of summation of the effect of two images. We have also introduced an estimate of the strength of a view, called intensity contrast, and used it to estimate whether the application of a method decreases the strength of that view. We have performed a subjective experiment

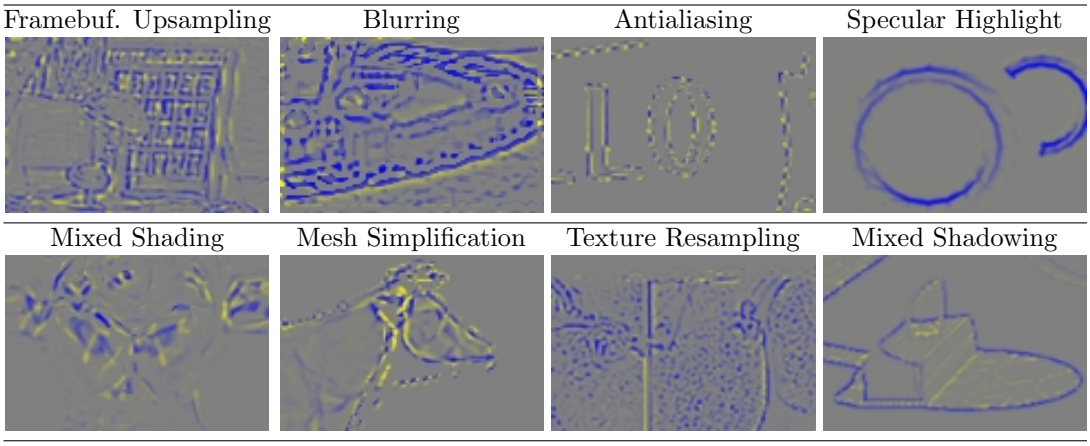


Figure 4.5: Intensity contrast changes due to selected methods. (Level 1 and level 3 are used for upsampling, blurring, mesh simplification, and texture resampling methods.) (From [20]. ©2010 Elsevier, reprinted with permission.)

on the selected methods.

We conclude that decreasing the rendering cost for one view may be an effective technique to increase the rendering performance of 3D stereoscopic content, while retaining the depth, quality, and sharpness of the original 3D rendering. The following methods provide an effective solution: framebuffer upsampling, blurring, specular highlight, mixed shading, texture resampling. On the other hand, mixed-level antialiasing, mesh simplification, and mixed shadowing, produce unacceptably low levels of quality and sharpness. The detailed experiment to evaluate the proposed method is presented in Section 5.4.

Chapter 5

Evaluation

This chapter is divided into three sections to present the experiments performed for each of the proposed methods. First two sections are related to the saliency calculation methods for animated meshes and animating objects and the third section is related to proposed stereo-rendering optimization technique.

5.1 Per-Vertex Saliency Model

5.1.1 Experiment Design

To verify the PVS model, we performed an experiment in which our aim was to compare the calculated saliency maps to the actual regions that are looked at by the users. For this purpose, we used a Tobii 1750 eye-tracker. In the experiment, three short video sequences (Figure 5.1) were shown to 12 subjects and the points that were looked at were captured. The duration of each animation was approximately 15s. The subjects had reported normal or corrected to normal vision and they freely viewed the animations with no assigned task. In order to evaluate our system, we followed the steps shown below:

1. We extracted the saliency maps for the animations that are used in the

experiment.

2. For each user, using the eye-tracking results, we marked the points of the animations that are looked at.

3. For each frame of the animation;

   (a) We calculated average saliency of all visible points; call it *average saliency*.

   (b) We calculated average saliency of all points that are marked as looked at, and compared this to *average saliency*.

5.1.2 Results and Discussion

There were several limitations in this experiment. Firstly, there was a margin of error, which was quite large considering that our saliency computation is performed over vertices. In order to tolerate this error, when calculating the average saliency of the points we used a small circular neighborhood of the points that were looked at. The radius of this circle was approximately 5% of the visible region. While this approximation tolerated the error to some extent, it caused the calculated average saliency of the points that were looked at to be closer to the average saliency of all visible points.

Another limitation was that there was a delay between a motion and the users response to that motion. To tolerate this error, we took the time of eye-tracker backwards by 0.4s, to take into account the subjects' reaction delay due to their perceptual processing of the shown animation.

Despite these limitations, the eye tracking results show that the subjects looked at the regions with significantly higher overall saliency than average. In the top row of Figure 5.2, saliency histograms for each saliency range are shown; whereas the bottom row shows the cumulative distribution of saliency values. Note that the saliency of visible points in these animations are scaled to [0,100] range in order to be able to compare the results. In these plots, the average

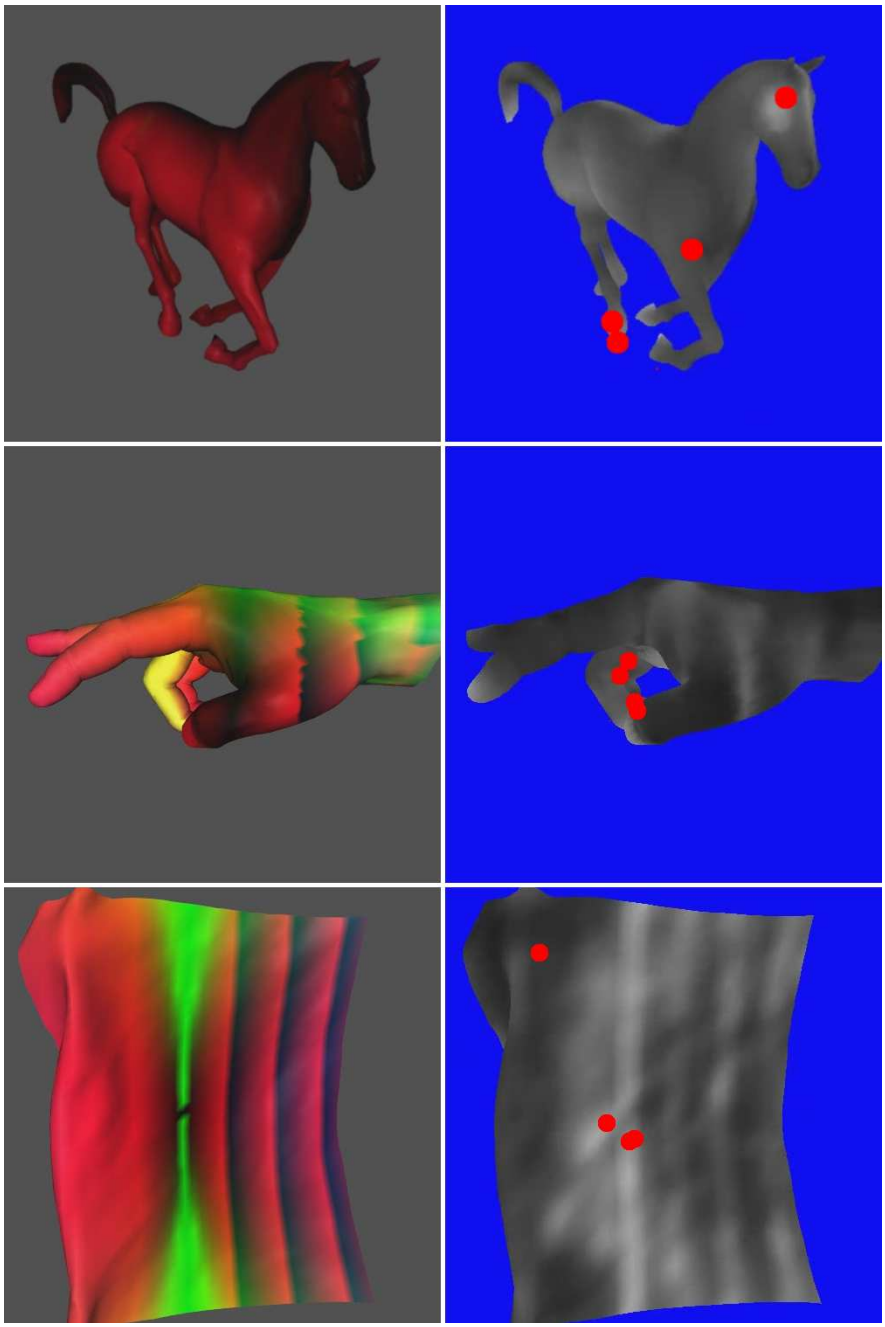


Figure 5.1: Samples from the three animation sequences used in the experiment. Left: original frames, right: saliency maps of the frames on the right (red dots indicate the regions that are looked at by the subjects for that frame). (From [21]. ©2010 ACM, reprinted with permission.)

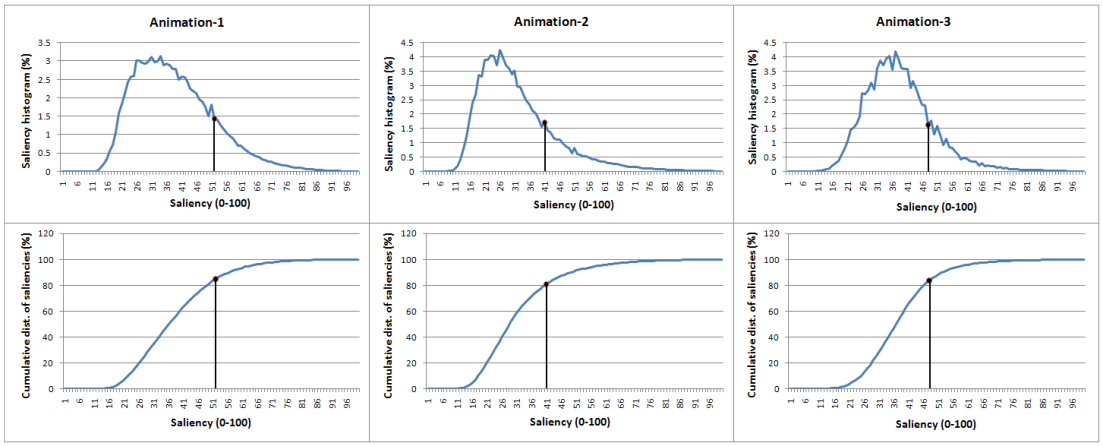


Figure 5.2: The results for the animation sequences used in the experiments. Plots at the top row show the average saliency histogram for each animation and the plots at the bottom show the cumulative distribution of saliency values. In each plot, the black dot indicates the average saliencies of the points that are looked at. (From [21]. ©2010 ACM, reprinted with permission.)

saliency values of all points that are the output of eye tracker are also shown. As shown in Figure 5.2, the average saliency of regions that the subjects look at for animations 1, 2, and 3 are ranked in the top 17%, 21%, and 19% of all visible points, respectively.

Furthermore, we tested the validity of the proposed saliency approach as follows. In addition to the actual users, we assumed 100 virtual users who look at random screen positions. Then, we compared the calculated average saliencies of the regions that the actual subjects look at to the regions that the virtual users looked at. In this comparison, we did not consider the points that were not on the animated models. Figure 5.3 shows that the actual users look at more salient regions compared to the virtual users and the difference is statistically significant ($p < 0.05$) according to the applied t-test.

These results show that our saliency metric identifies the regions that are likely to be looked at and determines the important parts of a 3D mesh. Moreover, these results can be considered as the worst case due to the limitations explained before.

Despite these limitations, the points that are looked at have significantly higher saliency values ($p < 0.05$) than the average saliency value of all visible

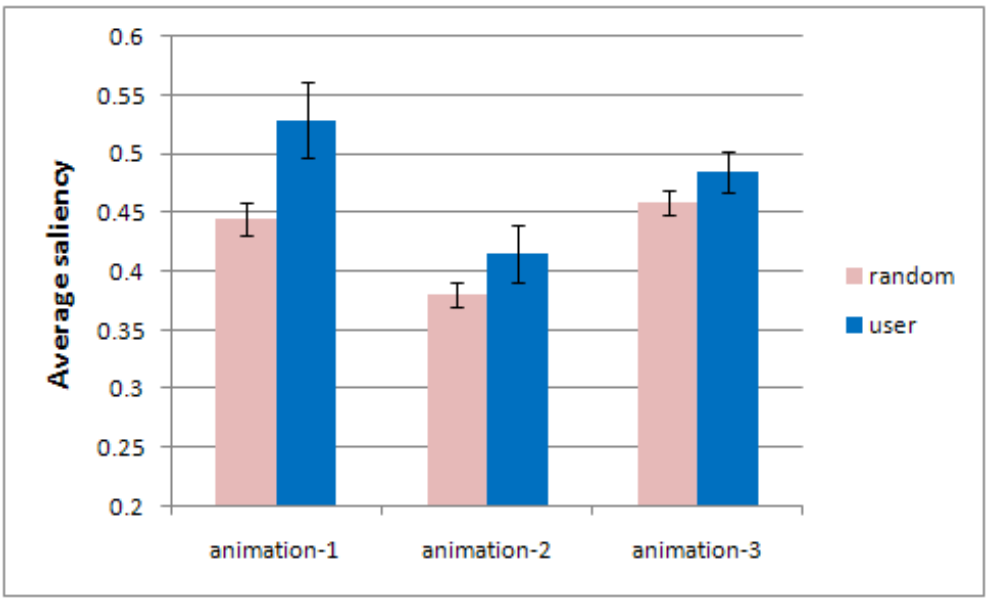


Figure 5.3: Comparison between the calculated average saliencies of the regions that are looked at by the actual users, and the randomly generated virtual users. Error bars indicate 95% confidence intervals. (From [21]. ©2010 ACM, reprinted with permission.)

points according to the applied t-test. Figure 5.2 shows the average saliency values of the points that are looked at by the users, against the average saliency values of all visible points through the animations. The average saliency of visible points are set to 1 in order to make the plots comparable.

We also calculated the correlation between the calculated saliency values and gaze points. For this purpose, we generated two video sequences for the contents used in the experiment. In the first video, the contents are colored according to the calculated saliency values. In the second video, the content is colored according to the eye tracker results such that a region that is looked at more is brighter. Then, we calculated the Pearson Correlation for each frame of the two video sequences. The average correlation results are shown in Table 5.1. Number of trials significantly effects correlation results because there are only a few fixation points for each frame of the animation which cannot be used to form a correct fixation map for a model. Also, in the ideal case, even if all subjects look at the most salient region, we don't get a high correlation value. Despite these problems, having a positive correlation value mimics that the regions having

higher saliency values are more likely to be the fixation points.

	correlation
animation-1	0.094
animation-2	0.191
animation-3	0.245

Table 5.1: Average correlation: saliency vs. fixations

5.2 Per-Object Saliency Model

5.2.1 Experiment Design

We performed a formal user study to validate the proposed POS model to calculate saliency of animating multiple objects. In the experiment, subjects looked at a 22" LCD display where twenty spheres animate for two minutes in a 3D room (Fig. 5.4). For each half-second of animation, a random object changed its motion state to another random state. The subjects' task was pressing a button when the color of a sphere becomes the target color which was shown to the subjects during the experiment. Interval time for two color change was at least three seconds. In order to avoid color or shape related bias, all spheres have the same size and randomly selected colors having close luminances. The reason for selecting different colors for each sphere is to decrease the pop-out effect for the spheres that change color. During an animation, we had three cases of color changes. The first case was the color change of highly salient spheres, which were chosen with strong decisions of our model. In second case, among spheres having the lowest saliencies, the most distant spheres to the previously calculated gaze points were chosen. In third case, we choosed the sphere in a fully random fashion. All cases were shown to the users in a mixed manner multiple times.

16 voluntary graduate or undergraduate level students (4 females, 12 males) whose average age was 23.75 attended to our experiment. All subjects had reported to have normal or corrected to normal vision and they were not informed

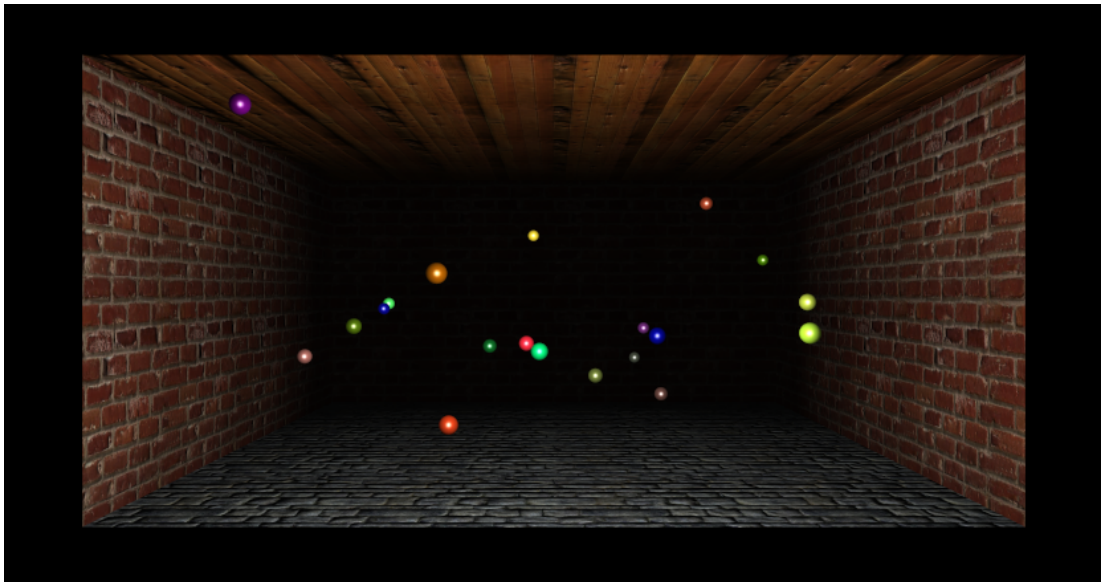


Figure 5.4: Sample screenshot from the experiment. (From [11]. ©2011 Springer, reprinted with permission.)

about the purpose of the experiment. The experiment was introduced to the subjects as a game. In the game, to get a higher score, they were told to press the button as soon as possible when an object changed its color to the target color. Before starting the experiment, each subject performed a trial case to learn how to play the game.

5.2.2 Results and Discussion

The results of the experiment can be seen in Fig. 5.5. We expect the response times to color changes of salient spheres to be shorter since they are expected to occur on the focus of attention. As expected, observers responded to color changes of salient spheres in a shorter time compared to those appeared on lowly salient spheres and randomly selected spheres. The differences for both cases are statistically significant ($p < 0.05$) according to the applied paired t-test.

For motion saliency in computer animations we analyzed psychological findings on the subject by conducting an eye-tracking experiment and developed a decision theoretic approach to momentarily determine perceptually significant

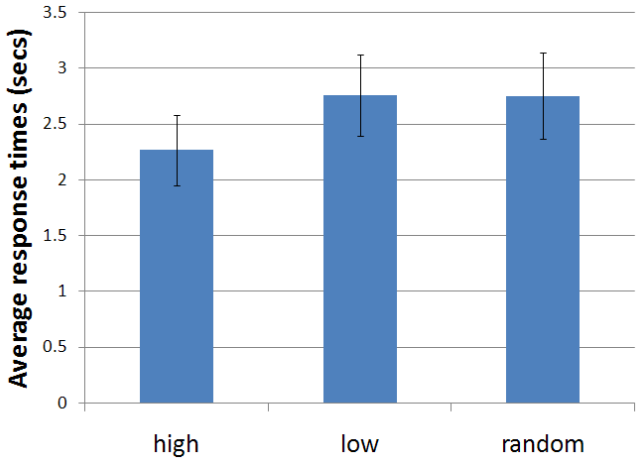


Figure 5.5: The results of the experiment. *High*, *low*, and *random* stand for the compared cases in which color changes are applied to objects having high, low, and random saliencies, respectively; according to the POS model. Error bars show the 95% confidence intervals. (From [20]. ©2011 Springer, reprinted with permission.)

objects. Our observations from the experiment showed that each phase of the motion has a different impact on perception. To that end, we defined six motion states and assigned attention values to them by considering their dominance. Individual attention values were determined for each object by considering their current state and elapsed time after a state initialization to include the impact of IOR. We elaborated our model by including the relationships of the objects on the scene with each other using Gestalt principles. Overall model makes decisions for the most salient object and its position is predicted as the gaze point. We carried-out a final experiment to evaluate the effectiveness of decisions our model make.

One of the limitations of our approach is the consideration of only bottom-up and stimulus-driven attention like most of other saliency works. Our final experiment had promising results although it included a simple task for users to search. On the other hand, more complex tasks could remove the effect of motion saliency. The model should be evaluated with further experiments and improved by the impact of new task-dependent cases. Furthermore, motion saliency and other object based saliency methods considering object material and shape should be compared. In our model we only consider saliency caused by motion. Dominancy

of other saliency parameters on motion saliency is not evaluated here.

5.3 Extended Per-Vertex Saliency Model

In this section, we present the evaluation of the motion-based EPVR model that is an adaptation of the motion-based saliency model (POS model) presented in 3.2 to 3D animated meshes.

5.3.1 Experimental Design

To evaluate the cluster-based saliency calculation model proposed in Section 3.3 we used the eye tracking data from the experiment presented in Section 5.1. The proposed model was applicable to the same content and the previously recorded gaze points are reused to analyze the saliency levels at these regions. Two of the short animations are analyzed with the newly proposed model. These animations include material properties of meshes in addition to the animations. While it is not desirable to have material properties while evaluating motion-based saliency since material properties also affect users' attention, the experiment is closer to the real world conditions with the presence of material properties.

5.3.2 Results and Discussion

The results of our analysis show that the motion based saliency calculation could work well for 3D animated meshes when motion based saliency is not dominated by another channel of saliency like geometry. Figure 5.6 shows the results of the analysis. In this figure, average saliencies in the gaze points of users are compared to average saliencies of randomly selected gaze points. In the analysis we considered a small neighborhood of the gaze points for two reasons, the first is to tolerate the small accuracy error (0.5 degrees of the visual field) of the used eye-tracker; and the second is our ability to see a small neighborhood in sharp

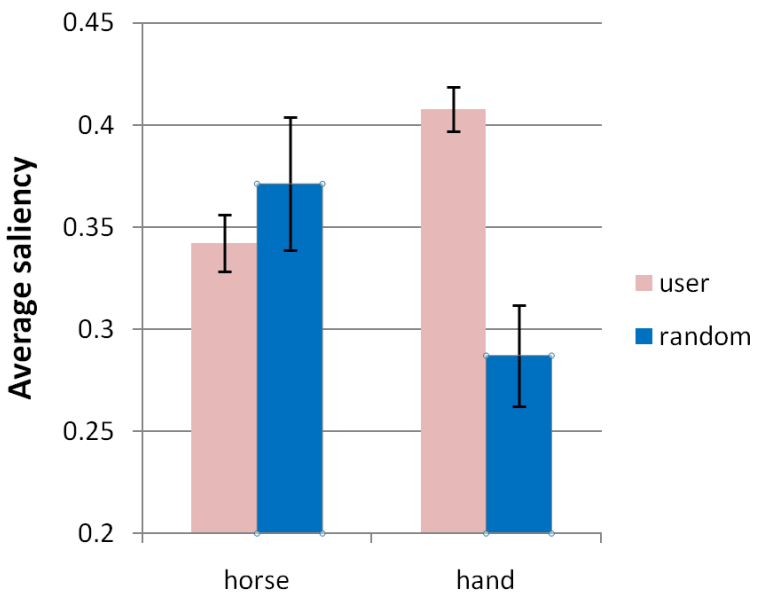


Figure 5.6: Experimental results for clustering-based saliency calculation. Error bars show the 95% confidence intervals.

detail (2 degrees around the gaze point).

As Figure 5.6 indicates, for different 3D models, the success of the clustering-based saliency calculation model differs. The proposed model is found to be very successful for the hand model such that the average saliency values of the users' gaze points are approximately 0.41 while for the random case it is around 0.28 in a scale where the vertex with the highest saliency value is set to 1 for each frame. However, for the horse model, a static part of the model, head, gathered the most attention resulting in less gazes on the motion-wise salient parts. There was no significant effect of motion saliency on reflecting the gaze points of the users. On the other hand, for the hand model, the proposed model reflected the possible attention centers well, possibly due to the absence of an outstanding distractor like head of the horse model. As the figure illustrates the error bars do not overlap and the results are found to be statistically significant according to the applied t-test.

Ref vs. Test	Ref vs. Test	Ref vs. Test
1-1 vs 1-2	2-2 vs 1-2	3-3 vs 1-2
1-1 vs 1-3	2-2 vs 1-3	3-3 vs 1-3
1-1 vs 1-4	2-2 vs 1-4	3-3 vs 1-4

Table 5.2: Test cases for scalable methods. (From [20]. ©2010 Elsevier, reprinted with permission.)

5.4 Attention-based Stereo Rendering Optimization

We implemented the proposed methods described in Section 4.1.1, and performed a formal experiment to observe whether the use of each method is perceptible. Our main approach is to compare differences of two cases: One of the cases is altering both left and right views in the same scale, and the other case is altering only one of the views with a different scale, using the methods in Table 4.1.

For the methods that can be applied in different scales, the first level is not modifying, and the fourth level is applying the method with the greatest strength. The actual correspondence of the scales for each method is shown in Table 4.1. For these methods, comparison cases are shown in Table 5.2. In the table, each cell corresponds to a comparison case. For instance, 1-1 vs 1-2 stands for the comparison of reference content in which the method is applied with level 1 for both views, with the test content in which the method is applied with level 1 for one view and with level 2 for the other view. Thus, there are 9 cases in total for each scalable method.

For the methods with two levels, there are two options: applying the method or not. The details of this type of methods are also shown in Table 4.1, and the comparison cases for these methods are shown in Table 5.3. In Table 5.3: 'on' stands for applying the method and 'off' stands for not applying the method. On the right side of the table, levels for the mixed shading method are shown.

Ref vs. Test	Ref vs. Test
on-on vs on-off	phong-phong vs phong-gouraud
off-off vs on-off	gouraud-gouraud vs phong-gouraud

Table 5.3: Test cases for non-scalable methods. (From [20]. ©2010 Elsevier, reprinted with permission.)

5.4.1 Experiment Design

5.4.1.1 Subjects

We recruited 61 subjects: 47 males and 14 females with a mean age of 24.6. The subjects were among voluntary undergraduate and graduate students with computer science background; and most of them do not have previous experience on rendering on stereoscopic displays. The subjects were not informed about the purpose of the experiment. All have self-reported normal or corrected vision.

5.4.1.2 Display

We used a Sharp Actius AL3DU stereoscopic laptop which has an NVIDIA GeForce Go 6600 graphics processor and a 15-inch XGA (1024×768) TFT 3D LCD display. In this display, 3D effect is provided by the parallax barrier technology. 3D perception is available for a single viewer, in a limited view angle and in a limited distance.

5.4.1.3 Procedure

For the experiment design, we followed the double-stimulus continuous-quality scale (DSCQS) method described in [2]. According to this procedure, subjects were shown a content, either test or reference, for about ten seconds; after a three seconds break, they were shown the other content. Then, both contents were shown for the second time, to obtain the subjective evaluations. The order of the reference and the test contents was determined randomly and subjects did not know whether they see the reference or the test content first. This process is

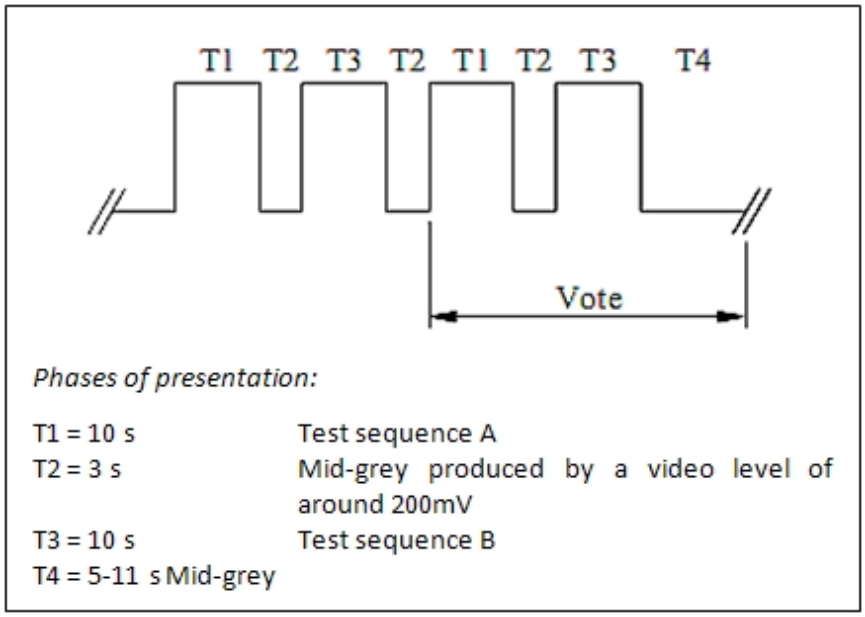


Figure 5.7: Presentation of test material. (From [20]. ©2010 Elsevier, reprinted with permission.)

Figure 5.8: Rating scales used for subjective assessments. (From [20]. ©2010 Elsevier, reprinted with permission.)

illustrated in Figure 5.7.

Our rating scale was also consistent with the scale that is recommended in ITU-R 500. A screenshot from our experiment system is shown in Figure 5.8. The rating scale is continuous and red pixels show the subject's rating for the corresponding field.

In the experiment, there are four methods with nine comparison cases and four methods with two comparison cases (Tables 4.1, 5.2, and 5.3). Thus, in total, there are 44 ($= 9 \times 4 + 2 \times 4$) different cases to be evaluated. Either 10 different videos or 25 different images were used per case on average. Each case is tested 15-20 times in total. For this purpose, each subject performed the experiment

for about 30 minutes, including a training case at the beginning and a 5 minute break in the middle of the experiment.

5.4.1.4 Assessment of Contents

Subjects evaluated both test and reference contents of all the cases separately, with respect to four criteria, as shown in Figure 5.8. The meaning of each criterion was explained to the viewers before the experiment begins. The motivation behind selecting these grading criteria is as follows:

- Quality: The primary goal of the experiment is to compare the quality of test and reference pairs. Quality denotes the perceived overall visual quality of the shown content. The labels "Excellent", "Good", "Fair", "Poor", and "Bad" were displayed alongside the scale.

- Depth: This criterion measures the apparent depth as reported by the user. Since stereoscopic displays are most beneficial for providing a better sense of depth, the effect of various modifications on apparent depth should be taken into account. For instance; if any modification that seems to retain quality causes a considerable amount of decrease in depth perception, the proposed optimization approach would be ineffective.

- Sharpness: This criterion is the subjective clarity of the details in an image, which is an important factor while evaluating graphical or image-based contents [113]. Sharpness has also been reported to be well correlated with quality of the contents [114], [24].

- Comfort: This criterion measures how distracting the scene is to the users. The visual comfort of a stereoscopic content may be affected by a number of factors such as left/right image misalignment, bad content creation, convergence-accommodation conflict and difficulty of getting the correct viewing position [69]. Therefore, the perceived comfort of a stereoscopic content is also reported as important and widely-used as a criterion in subjective experiments for stereoscopic displays [69], [131]. Similar to depth

criterion, the resulting comfort of an applied method should be taken into consideration to find out whether it is affected by our optimization approach.

5.4.2 Results and Discussion

To determine the difference between the reference and test content, the Test minus Reference score was used. A score of zero means that the Test sequence was rated equivalently to the reference sequence, and a negative score means that the test content was rated lower than the reference content. Error bars in the figures below show the 95% confidence interval of the mean, which corresponds to the range within which the mean is expected to fall with 95% certainty. Data points in the non-overlapping error bars indicate statistical difference at the $p < 0.05$ level. For each of the 44 test cases, a paired samples t-test was also applied; with $p < 0.05$ level to represent statistically significant difference between the pairs.

The parallax-barrier display that was used in the experiment has a narrow range of correct viewing position. Occasionally, this may cause a difficulty in getting the proper viewing position and leads to abnormal ratings for the contents that are seen from the wrong viewpoint. Therefore, it is likely to have outliers among the experimental results. The outliers were detected as the ratings that lie outside the region of $mean \pm 2 \times stddev$ [109], and the cases reported by subjects.

5.4.2.1 Framebuffer Upsampling

Figure 5.9 shows the experimental results of framebuffer upsampling method. The quality and sharpness results show that all mixed pairs (1-2, 1-3, and 1-4) are perceived better than 2-2 pair, without any loss of apparent depth and comfort. The differences in sharpness are even larger for 3-3 pair, which is consistent with the hypothesis. Furthermore, the 1-2 pair is close to 1-1 pair in all rating criteria. On the other hand, the 1-1 pair statistically differs from 1-3 and 1-4 pairs, which may be the result of the Lanczos algorithm that was used for upsampling.

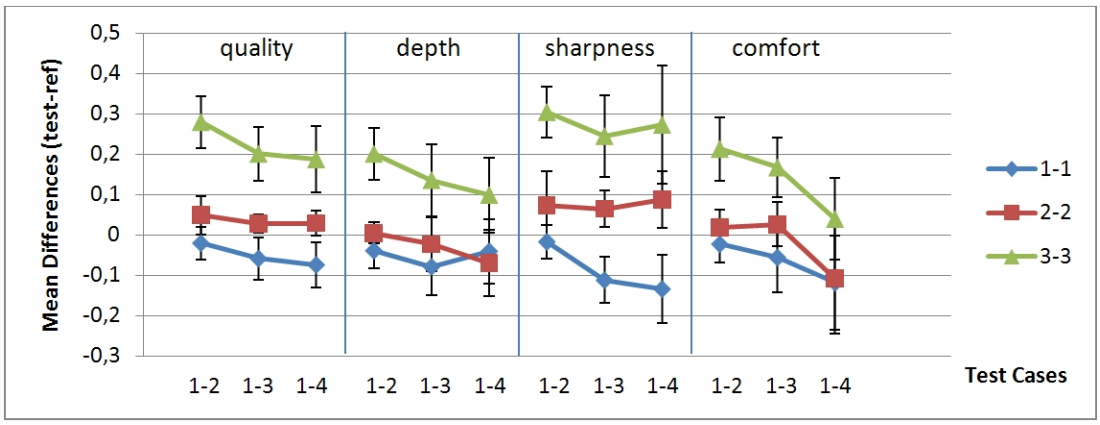


Figure 5.9: Experimental results for framebuffer upsampling method (Error bars show the 95% confidence interval of the mean). (From [20]. ©2010 Elsevier, reprinted with permission.)

Therefore, other upsampling algorithms, such as B-Spline based methods, may fit better to higher level of simplification; since they result in upsampled images that have lower intensity contrast (Figure 5.10).

This pattern of results suggests that memory and computation savings can be achieved with application of the framebuffer upsampling method in Level 2 to one view, with equivalent perceived quality and sharpness. Moreover, the cost of advanced rendering techniques, such as real-time ray tracing, can be decreased for one view by using framebuffer upsampling.

5.4.2.2 Blurring

Figure 5.11 shows that blurring the right-eye view in Level 2 or Level 3 had no perceived effect in all rating criteria. On the other hand, there is a statistically significant difference between 1-1 and 1-4 pairs. The series 2-2 and 3-3 in the quality panel indicates that when one of the views is not blurred in a pair, the perceived quality of this pair is higher than blurring both views at the same time.

In conclusion, the results are consistent with our expectation that the blurred image is suppressed by the original view, due to the decrease in the intensity contrast. The similarity in quality between 1-1 and 1-3 pairs and the significant

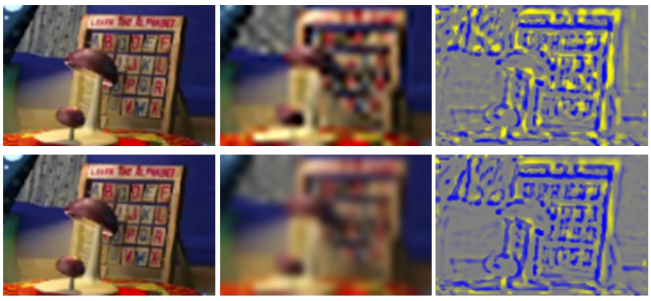


Figure 5.10: Comparison of Upsampling Algorithms. Top-left and Bottom-left: Original image, Top-middle: Level 2 image upsampled with Lanczos filter, Top-right: Intensity contrast change map of top-middle image (compared to the original image), Bottom-middle: Level 2 image upsampled with B-Spline filter, Bottom-right: Intensity contrast change map of bottom-middle image (compared to the original image). (From [20]. ©2010 Elsevier, reprinted with permission.)

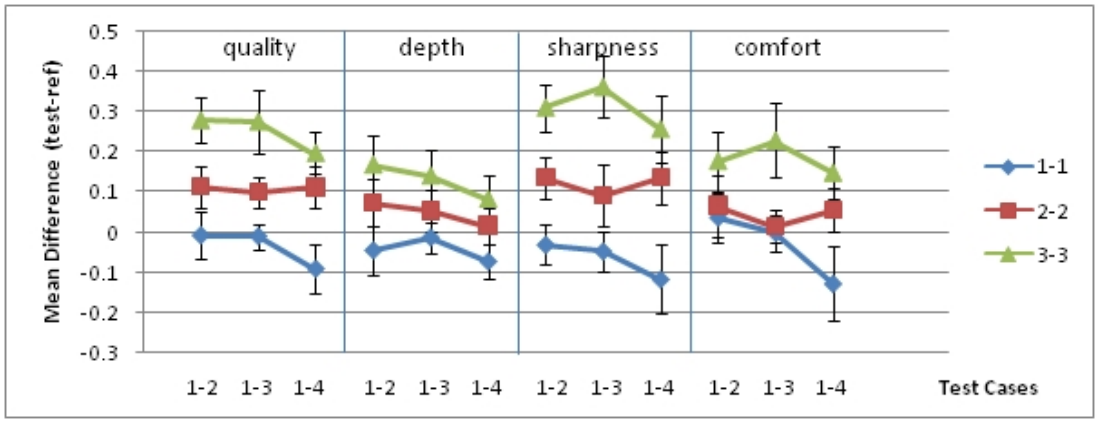


Figure 5.11: Experimental results for blurring method (Error bars show the 95% confidence interval of the mean). (From [20]. ©2010 Elsevier, reprinted with permission.)

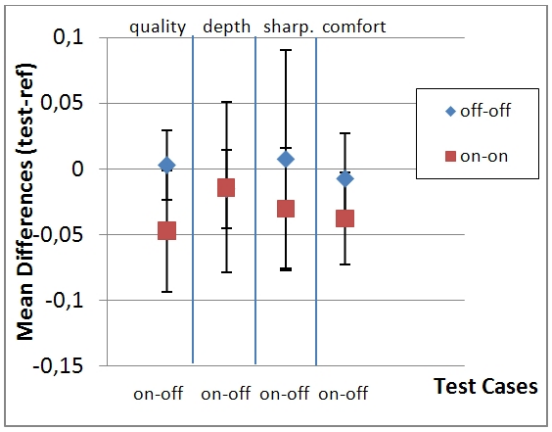


Figure 5.12: Experimental results for mixed-level antialiasing method (Error bars show the 95% confidence interval of the mean). (From [20]. ©2010 Elsevier, reprinted with permission.)

quality decrease from 1-3 to 3-3 pairs show that non-blurred content determines the perceived quality of the 1-3 pair, which strongly supports our idea. However, the level of blurring should be carefully considered, since exaggerating the blur effect may result in a poor depth perception and a low quality, as the results for 1-4 pair indicate.

5.4.2.3 Mixed-level Antialiasing

Figure 5.12 shows that the mixed pair for antialiasing was rated close in overall quality to the non-antialiased pair, whereas antialiasing both views show higher quality than the mixed pair. The results for perceived sharpness and comfort show a similar pattern to results of perceived quality. On the other hand, the results show that antialiasing has no effect on the apparent depth.

These results indicate that the antialiased image is suppressed by the non-antialiased image. This outcome is in accordance with our expectations, since antialiasing decreases the intensity contrast. Thus, turning off antialiasing in only one view is not appropriate, and should only be applied or disabled on both of the views.

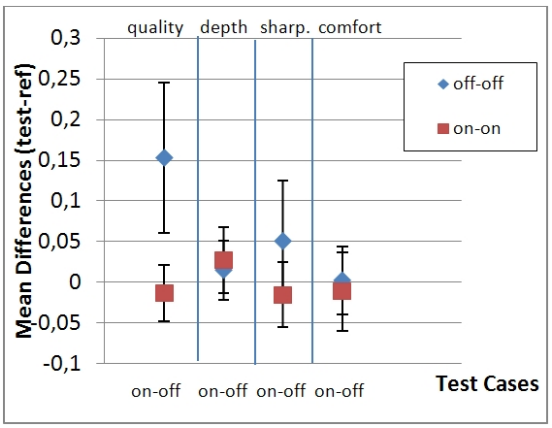


Figure 5.13: Experimental results for specular highlight method (Error bars show the 95% confidence interval of the mean). (From [20]. ©2010 Elsevier, reprinted with permission.)

5.4.2.4 Specular Highlight

The results for the specular highlight method are shown in Figure 5.13. It is apparent that the perceived quality, depth, sharpness, and comfort of the mixed pair were rated similarly to the reference pair, while the quality of the pair with specular highlight removed in both views was rated lower than the mixed pair. Therefore, removing specular highlight in one view in a stereo pair has a small effect on the overall stereo image and thus can be used for mixed stereo rendering.

5.4.2.5 Mixed Shading

Figure 5.14 shows that the mixed shading pair was rated with equivalent results to Phong-shaded pair, and with higher quality than the Gouraud shaded pair. The explanation of this result is that the intensity contrast is increased when Phong shading is used instead of Gouraud shading, because the specular highlight is more visible in Phong shading.

These results indicate that mixed shading provides a viable alternative for stereoscopic rendering. Nevertheless, the situation will not be probably the same when "extreme" shading methods, such as flat shading, are used for one view.

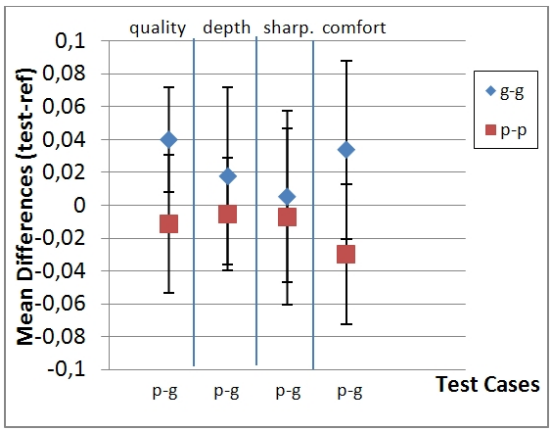


Figure 5.14: Experimental results for mixed shading method (Error bars show the 95% confidence interval of the mean). (From [20]. ©2010 Elsevier, reprinted with permission.)

Since flat shading increases the intensity contrast by resulting in a color discontinuity on the edges, flat shaded view will be dominant against the Phong shaded view on the edges.

5.4.2.6 Mesh Simplification

The results for the mesh simplification method are shown in Figure 5.15. It is not possible to obtain a general inference by looking at the quality, depth, sharpness, and comfort results. This situation is not in conflict with our prediction, which is based on the idea that intensity contrast is higher on the edges (especially on the silhouette edges) and each mesh will probably dominate on its own edges. Therefore, the perceived 3D mesh is likely not one of the meshes, but an unpredictable combination of them.

As a result, it is not easy to predict the effect of a mixed pair on the stereo image, while using meshes of different level of detail for each view is not appropriate. However, simplifying the mesh for a single view may be applicable if the silhouette is preserved. One possible improvement may be application of a silhouette-preserving mesh simplification method which does not cause a significant increase in the intensity contrast of the simplified mesh.

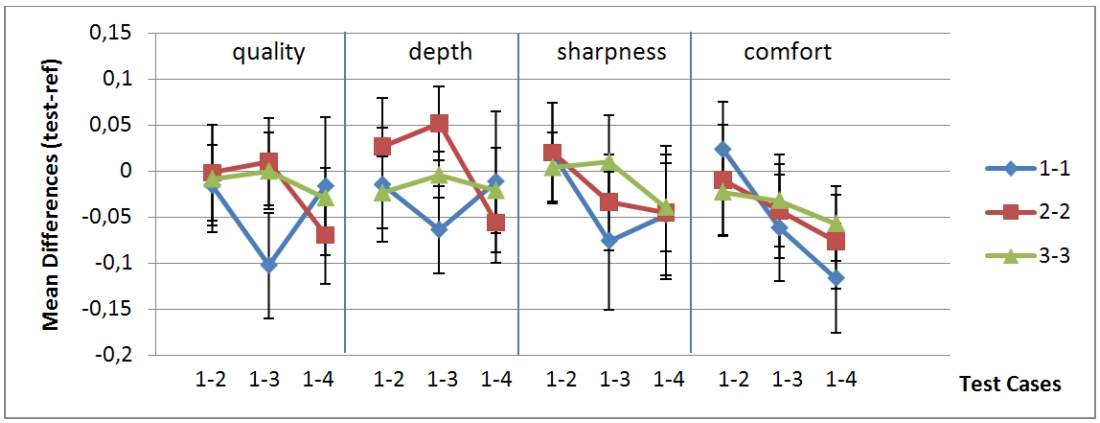


Figure 5.15: Experimental results for mesh simplification method (Error bars show the 95% confidence interval of the mean). (From [20]. ©2010 Elsevier, reprinted with permission.)

5.4.2.7 Texture Resampling

Figure 5.16 shows that the quality responses of the 1-2 and 1-3 pairs were close to the original 1-1 pair. All the mixed pairs were rated higher than the 3-3 pair, both for quality and sharpness. These results meet our expectations. On the other hand, the ratings for the 2-2 pair contradict our predictions. Our expectation was that the 2-2 pair would be rated lower than the mixed pairs, for which the quality is determined by the Level 1 view according to our hypothesis. A likely explanation of this contradictory situation is as follows: In our experiment, we observed that the resolution of the Level 2 texture maps were already sufficient for our objects since the area to cover is smaller than the size of the Level 2 texture maps; so that the Level 1 texture maps cannot provide higher quality than the Level 2 texture maps. In this regard, if we consider the Level 1 and Level 2 texture maps as similar in final rendered image, the results seem to be consistent with our expectation. Thus, the texture resampling method provides a viable solution for stereoscopic rendering.

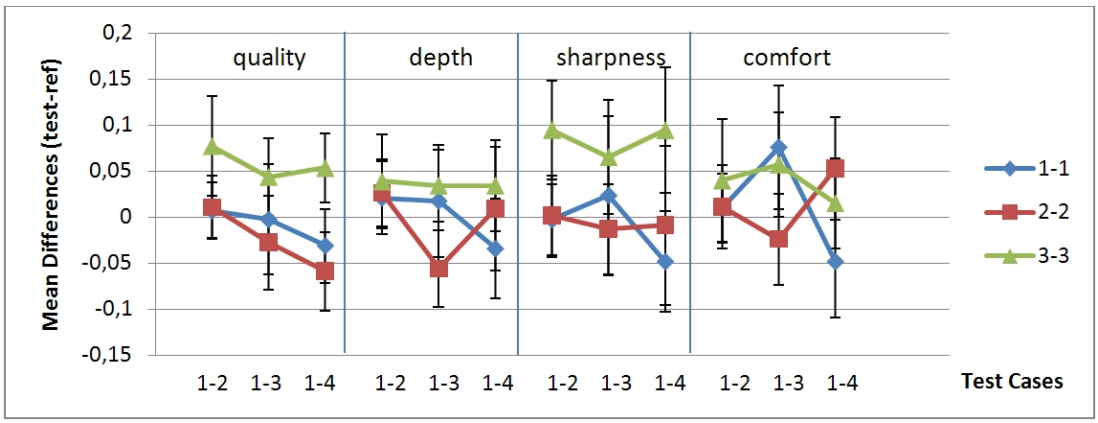


Figure 5.16: Experimental results for texture resampling method (Error bars show the 95% confidence interval of the mean). (From [20]. ©2010 Elsevier, reprinted with permission.)

5.4.2.8 Mixed Shadowing

Figure 5.17 illustrates that the mixed pair has closer ratings for quality to the case in which both views are not shadowed. Furthermore, the original reference pair has significantly higher quality ratings than the mixed pair. These two results imply that using shadows in only one view is not a feasible solution, as it affects perceived quality.

Another result inferred from the sharpness ratings is that shadowed (on-on) and shadowless (off-off) reference pairs are sharper than the mixed pair, and the difference is more apparent while comparing the shadowed reference pair to the mixed pair. This situation may be explained as the following: In a mixed pair, right and left views may become dominant on different regions and this decreases the sharpness of the perceived stimulus.

Consequently, using shadow for a single view is not appropriate since it does not increase the quality and depth perception to a higher level than the pair without shadows. A mixed pair is rated to have lower comfort and sharpness than the reference pairs.

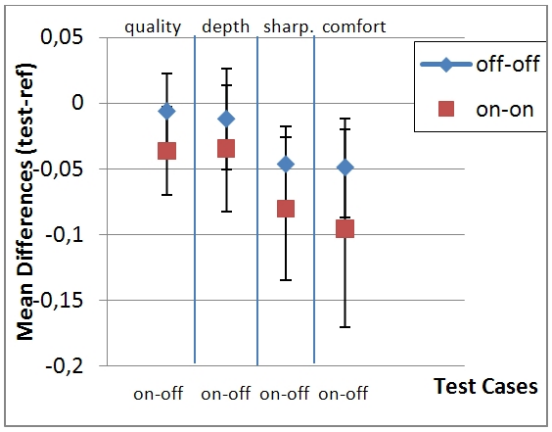


Figure 5.17: Experimental results for mixed shadowing method (Error bars show the 95% confidence interval of the mean). (From [20]. ©2010 Elsevier, reprinted with permission.)

5.4.2.9 General Discussion

Table 5.4 summarizes the feasibility of using the selected methods for mixed stereoscopic rendering, thus decreasing rendering complexity. Our experiment results show that it is possible to decrease the rendering cost of a 3D frame using methods: framebuffer upsampling, blurring, specular highlight, mixed shading, texture resampling. However, this optimization approach is not feasible for effects such as mixed-level antialiasing and mixed shadowing.

Our hypothesis suggests that using different stimuli for each eye can be used for optimization purposes if the applied effect decreases the intensity contrast and as a result the high-quality view dominates in the mixed pair. In the meantime, one important point to consider is that the difference in levels, between the two views should not be increased significantly. For example, higher levels of blurring one view decreases the perceived quality and depth, as the experiment results have shown.

Method	**Expectation**	**Applying to single view**
Framebuffer Upsampling	Upsampled view is suppressed. (Not as apparent as the blurring case)	Feasible
Blurring	Blurred view is suppressed.	Feasible
Mixed-level Antialiasing	Antialised view is suppressed.	Not feasible
Specular highlight	The view with specular highlight suppresses the other.	Feasible
Mixed Shading	Shaded with Phong model suppresses the Gouraud in general.	Feasible
Mesh Simplification	Two meshes may not be perceived as a single mesh. Not appropriate to use.	Not feasible
Texture Resampling	Texture mapped with higher resolution image suppresses the other.	Feasible
Mixed Shadowing	Silhouette of shadows become apparent but may result in discomfort since brighter parts suppress inside the shadowed regions.	Not feasible

Table 5.4: Summary of the experiment. (From [20]. ©2010 Elsevier, reprinted with permission.)

Chapter 6

Conclusion

Computer Graphics is a discipline in which the main purpose is generating computer generated scenes that are in good quality. Users could assess the quality of the rendered scenes merely based on their perception of these scenes. In this thesis, we utilized a portion of the significant perceptual knowledge for computer graphics.

Our approach is composed of two main parts: In the first part, our main focus is to find out the perceptually attractive regions of the scenes and in the second part our main focus is to utilize the binocular vision mechanism to optimize stereoscopic rendering without sacrificing visual quality. Apart from the proposed perceptual techniques, a detailed interdisciplinary literature survey combining areas in Psychology, Visual Perception, Neuroscience, and Computer Graphics is presented in Chapter 2. Although we have covered the most important literature for our research in detail, there is a notable perception knowledge that could be utilized in computer graphics which is practically impossible to fit in a thesis.

We have proposed saliency calculation techniques for two type of contents. First one, PVS, works on single 3D meshes and tries to find the salient regions of them due to their geometry, animation, and material properties. The second one, POS, works on multiple objects that are animating and its purpose is to find out the most salient object in terms of its motion for each frame of the

animation. We have evaluated both of these studies and statistically verified that the proposed metrics are successful. The models proposed in this thesis are based on the bottom-up part of visual attention. A possible direction for future research could be incorporating the task-based, top-down visual attention into saliency computation framework. As explained in Section 2.1.1.3, task based attention greatly affects our sensitivity to the visual scene. After a saliency analysis as a pre-process without the presence of a task, such an extension could work in real-time depending on the user's task at the moment.

Salient regions could be considered as the important parts of visual scene and saliency information has a great potential for utilizing in Computer Graphics. It could be used for perceptual level of detail adjustment, perceptual camera control, and for artistic purposes like automatized caricaturization and cubist like rendering frameworks. Another, application area could be games in which difficulty level could be adjusted perceptually via, for instance, placing a search target into a salient or non-salient area.

There are many attempts to build a perceptual metric to evaluate quality of 3D models; however, to the best of our knowledge, none of them uses attention principles in their assessments. Since we are more sensitive to the salient regions of 3D models, it could be useful to employ saliency information in a perceptual quality metric.

We have also investigated the graphical methods that could be used for stereoscopic rendering optimization utilizing the binocular suppression theory of binocular vision. We have performed a detailed experiment and reached this general conclusion: A modification which raises the intensity contrast does not require application to both views. On the other hand, if the intensity contrast of the modified view is lower than the original image, then the optimized pair provides the same percept as the result of traditional rendering. To be more specific, the proposed stereoscopic rendering optimization technique is applicable for several graphical methods e.g., framebuffer upsampling, specular reflection, blurring, and shading.

Our general finding about the applicability of graphical methods could be investigated in more detail for specific graphical methods like specular reflection, use of shadows etc. Moreover, the proposed rendering optimization is for rendering only two views (left and right views), autostereoscopic multiview displays have much more views to be rendered concurrently and how to apply the proposed optimization strategy for such displays is also a possible research possibility for the future.

Bibliography

[1] ITU recommendation p.910: Subjective video quality assessment methods for multimedia applications, 1996.

[2] ITU-r recommendation bt.500-11: Methodology for the subjective assessment of the quality of television pictures, 2002.

[3] R. A. Abrams and S. E. Christ. Motion onset captures attention. *Psychological Science*, 14:427–432, 2003.

[4] S. Adelson and L. Hodges. Stereoscopic ray tracing. *The Visual Computer*, 10(3):127–144, 1993.

[5] T. Akenine-Moller, E. Haines, and N. Hoffman. *Texturing*, pages 157–163. AK Press, Wellesley, 3rd edition, 2008.

[6] T. Akenine-Moller, E. Haines, and N. Hoffman. *Visual Appearance*, pages 124–133. AK Peters, Wellesley, 3rd edition, 2008.

[7] S. Albin, G. Rougeron, B. Peroche, and A. Tremeau. Quality image metrics for synthetic images based on perceptual color differences. *IEEE Transactions on Image Processing*, 11(9):961 – 971, Sept. 2002.

[8] R. A. Andersen. Neural mechanisms of visual motion perception in primates. *Neuron*, 18:865–872, 1997.

[9] R. Arnheim. *Art and Visual Perception-A Psychology of the Creative Eye*. University of California Press, 1954.

[10] S. Arpa. A perceptual approach for cubist style rendering. Master's thesis, Bilkent University, 2012.

[11] S. Arpa, A. Bulbul, and T. Capin. A decision theoretic approach to motion saliency in computer animations. In J. Allbeck and P. Faloutsos, editors, *Motion in Games*, volume 7060 of *Lecture Notes in Computer Science*, pages 168–179. Springer Berlin / Heidelberg, 2011.

[12] H. Asher. Suppression theory of binocular vision. *Br. J. Opthalmol.*, 37:37–49, 1953.

[13] N. Aspert, D. Santa-cruz, and T. Ebrahimi. Mesh: Measuring errors between surfaces using the hausdorff distance. In *Proceedings of 2002 IEEE International Conference on Multimedia and Expo, ICME '02*, volume 1, pages 705–708, 2002.

[14] A. Berthold. The influence of blur on the perceived quality and sensation of depth of 2d and stereo images. Technical report, Kyoto, 1997.

[15] 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*, 24:65–75, January 2009.

[16] R. Blake and J. Camisa. The inhibitory nature of binocular rivalry suppression. *J. Exp. Psychol*, 5:315–323, 1979.

[17] R. Blake and N. K. Logothetis. Visual competition. *Nat. Rev. Neurosci.*, 3(1):13–21, 2002.

[18] 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, SIGGRAPH '98*, pages 299–309, New York, NY, USA, 1998. ACM.

[19] A. Bulbul, T. Capin, G. Lavoue, and M. Preda. Assessing visual quality of 3-d polygonal models. *Signal Processing Magazine, IEEE*, 28(6):80–90, Nov. 2011.

[20] A. Bulbul, Z. Cipiloglu, and T. Capin. A perceptual approach for stereoscopic rendering optimization. *Computers & Graphics*, 34(2):145 – 157, 2010.

[21] A. Bulbul, C. Koca, T. Capin, and U. Güdükbay. Saliency for animated meshes with material properties. In *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, APGV '10, pages 81–88. ACM, 2010.

[22] F. W. Campbell and J. G. Robson. Application of Fourier analysis to the visibility of gratings. *The Journal of Physiology*, 197(3):551–566, 1968.

[23] 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*, EGRW '03, pages 270–280, Aire-la-Ville, Switzerland, 2003. Eurographics Association.

[24] J. Caviedes and F. Oberti. A new sharpness metric based on local kurtosis, edge and energy information. *Signal Processing: Image Communication*, 19(2):147–161, 2004.

[25] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, 1998.

[26] G. Cimen, A. Bulbul, B. Ozguc, and T. Capin. Perceptual caricaturization of 3D models. In *Proceedings of 27th International Symposium on Computer and Information Sciences*, To appear.

[27] J. P. Collomosse and P. M. Hall. Cubist style rendering from photographs. *IEEE Transactions on Visualization and Computer Graphics*, 4:443–453, 2002.

[28] M. Corsini, E. Drelie Gelasca, T. Ebrahimi, and M. Barni. Watermarked 3D Mesh Quality Assessment. *IEEE Transactions on Multimedia*, 9(2):247–256, 2007.

[29] S. Daly. Digital images and human vision. chapter The visible differences predictor: an algorithm for the assessment of image fidelity, pages 179–206. MIT Press, Cambridge, MA, USA, 1993.

[30] S. Daly. Engineering observations from spatiovelocity and spatiotemporal visual models. In *IS&T/SPIE Conference on Human Vision and Electronic Imaging III. SPIE*, volume 3299, pages 180–191, 1998.

[31] E. Drelie Gelasca, T. Ebrahimi, M. Corsini, and M. Barni. Objective Evaluation of the Perceptual Quality of 3D Watermarking. In *Proceedings of IEEE International Conference on Image Processing, ICIP*, 2005.

[32] A. Es and V. Isler. GPU based real time stereoscopic ray tracing. In *Proceedings of 22nd International Symposium on Computer and Information Sciences*, pages 1–7, 2007.

[33] C. Fehn. Depth-image based rendering (DIBR), compression and transmission for a new approach on 3DTV. In *SPIE*, pages 93–104, 2004.

[34] M. Feixas, M. Sbert, and F. Gonzalez. A unified information-theoretic framework for viewpoint selection and mesh saliency. *ACM Transactions on Applied Perception*, 6(1):Article no. 1, 23 pages, 2009.

[35] 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, SIGGRAPH '97*, pages 143–152, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[36] D. A. Forsyth and J. Ponce. *Linear Filters and Convolution*, chapter 7, pages 135–140. Prentice Hall, New Jersey, 2003.

[37] J. P. Frisby and J. V. Stone. *Seeing and Psychophysics*, chapter 12, pages 281–306. The MIT Press, 2nd edition, 2010.

[38] J. P. Frisby and J. V. Stone. *Seeing Motion, Part I*, chapter 14, pages 324–353. The MIT Press, 2nd edition, 2010.

[39] S. Fu, H. Bao, and Q. Peng. Accelerated rendering algorithm for stereoscopic display. *Computers & Graphics*, 20(2):223–229, 1996.

[40] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '97*, pages 209–216, New York, NY, USA, 1997. ACM.

[41] H. Gouraud. Continuous shading of curved surfaces. *IEEE Transaction on Computer*, 20:623–629, 1971.

[42] C. Halit and T. K. Capin. Multiscale motion saliency for keyframe extraction from motion capture sequences. *Journal of Visualization and Computer Animation*, 22(1):3–14, 2011.

[43] M. Halle. Multiple viewpoint rendering. In *Proccedings of 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '99*, pages 243–254, New York, 1998. ACM.

[44] J. Hasselgren and T. Akenine-Moller. An efficient multi-view rasterization architecture. In *Eurographics Symposium on Rendering*, pages 61–72. Eurographics Assoc., 2006.

[45] T. He and A. Kaufman. Fast stereo volume rendering. *IEEE Visualization*, pages 49–57, 1996.

[46] D. Hearn and M. P. Baker. *Computer Graphics with OpenGL*, chapter 10, pages 563–576. Prentice Hall, New Jersey, 3rd edition, 2004.

[47] D. Hearn and M. P. Baker. *Computer Graphics with OpenGL*, chapter 10, pages 628–634. Prentice Hall, New Jersey, 3rd edition, 2004.

[48] E. Hering. *Outlines of a Theory of the Light Sense*. Harvard University Press, 1964.

[49] O. Hershler and S. Hochstein. At first sight: A high-level pop out effect for faces. *Vision Research*, 45(13):1707 – 1724, 2005.

[50] O. Hershler and S. Hochstein. With a careful look: Still no low-level confound to face pop-out. *Vision Research*, 46(18):3028 – 3035, 2006.

[51] A. P. Hillstrom and S. Yantis. Visual motion and attentional capture. *Perception & Psychophysics*, 55(4):399–411, Apr. 1994.

[52] D. Hoffman, A. Girshick, K. Akeley, and M. Banks. Vergence-accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of Vision*, 8(3):33, 2008.

[53] D. D. Hoffman and M. Singh. Salience of visual parts. *Cognition*, 63:29–78, 1997.

[54] M. Hollins and E. H. L. Leung. The influence of color on binocular rivalry. *Visual Psycophysics and Physiology*, pages 181–190, 1978.

[55] I. Howard and B. Rogers. *Seeing in Depth*. Oxford University Press, 2008.

[56] I. P. Howard and B. J. Rogers. *Binocular Fusion and Rivalry*. Oxford Univ. Press, New York, 1995.

[57] S. Howlett, J. Hamill, and C. O'Sullivan. An experimental approach to predicting saliency for simplified polygonal models. In *Proceedings of the 1st Symposium on Applied perception in graphics and visualization, APGV '04*, pages 57–64, New York, NY, USA, 2004. ACM.

[58] S. Howlett, J. Hamill, and C. O'Sullivan. Predicting and evaluating saliency for simplified polygonal models. *ACM Trans. Appl. Percept.*, 2(3):286–308, July 2005.

[59] L. Itti and C. Koch. A comparison of feature combination strategies for saliency-based visual attention systems. *Journal of Electronic Imaging*, 10:161–169, 1999.

[60] 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.

[61] L. Itti and C. Koch. Computational modeling of visual attention. *Nature Reviews Neuroscience*, pages 194–203, 2001.

[62] 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, 1998.

[63] A. Kalaiah and T. Capin. Unified rendering pipeline for autostereoscopic displays. In *Proceedings of 3DTV Conference*, pages 1–4, 2007.

[64] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00*, pages 279–286, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[65] D. H. Kelly. Motion and vision ii. stabilized spatio-temporal threshold surface. *Optical Society of America*, 69(10):1340–1349, 1979.

[66] Y. Kim. *Saliency-guided Graphics and Visualization*. PhD thesis, University of Maryland, College Park, 2008.

[67] Y. Kim and A. Varshney. Persuading visual attention through geometry. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):772–782, 2008.

[68] K. Koffka. *Principles of Gestalt Psychology*. Routledge, Lond. :, 1935.

[69] F. L. Kooi and A. Toet. Visual comfort of binocular and 3D displays. *Displays*, 25:99–108, 2004.

[70] 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, February 2009.

[71] G. Lavoué. A multiscale metric for 3d mesh visual quality assessment. *Computer Graphics Forum*, 30(5):1427–1437, 2011.

[72] 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, November 2010.

[73] G. Lavoué, E. D. Gelasca, F. Dupont, A. Baskurt, and T. Ebrahimi. Perceptually driven 3D distance metrics with application to watermarking. In *Proceedings of SPIE Applications of Digital Image Processing XXIX*, volume 6312, 2006.

[74] C. H. Lee, A. Varshney, and D. W. Jacobs. Mesh saliency. *ACM Transactions on Graphics (Proc. of SIGGRAPH'05)*, 24(3):659–666, 2005.

[75] S. Lee, G. J. Kim, and S. Choi. Real-time tracking of visually attended objects in interactive virtual environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 29–38, California, 2007.

[76] B. Li, G. W. Meyer, and R. V. Klassen. A comparison of two image quality models. In *Proceedings of SPIE Human Vision and Electronic Imaging III*, volume 3299, pages 98–109, 1998.

[77] P. Lindstrom and G. Turk. Image-driven simplification. *ACM Transactions on Graphics*, 19:204–241, July 2000.

[78] Y. S. Liu, M. Liu, D. Kihara, and K. Ramani. Salient critical points for meshes. In *Proceedings of ACM Symposium on Solid and Physical Modeling*, pages 277–282, 2007.

[79] P. Longhurst and A. Chalmers. User validation of image quality assessment algorithms. In *Proceedings of the Theory and Practice of Computer Graphics, TPCG'04*, pages 196–202, Washington, DC, USA, 2004. IEEE Computer Society.

[80] 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.

[81] J. Lubin. *A Visual Discrimination Model for Imaging System Design and Evaluation*. World Scientific, 1995.

[82] D. Luebke, B. Watson, J. D. Cohen, M. Reddy, and A. Varshney. *Level of Detail for 3D Graphics*. Elsevier Science Inc., New York, NY, USA, 2002.

[83] P. D. Luebke and B. Hallen. Perceptually-driven simplification for interactive rendering. In *Proceedings of Eurographics Workshop on Rendering Techniques*, pages 223–234, 2001.

[84] R. Mantiuk, K. Myszkowski, and H.-P. Seidel. Visible difference predicator for high dynamic range images. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 2763–2769, 2004.

[85] R. McDonnell, M. Larkin, B. Hernández, I. Rudomin, and C. O'Sullivan. Eye-catching crowds: saliency based selective variation. In *ACM SIGGRAPH papers*, SIGGRAPH '09, article no: 55, 10 pages, New York, NY, USA, 2009. ACM.

[86] B. Mendiburu. *Learning 3D Cinematography*, chapter 3, pages 35–46. Elsevier, 2009.

[87] B. Mendiburu. *Stereoscopic Vision and 3D Cinematography*, chapter 2, pages 11–34. Elsevier, 2009.

[88] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Proceedings of VisMath*, 2002.

[89] M. Mortara and M. Spagnuolo. Semantics-driven best view of 3D shapes. *Computers & Graphics*, 33(3):280–290, 2009.

[90] 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, SIGGRAPH '01*, pages 221–230, New York, NY, USA, 2001. ACM.

[91] V. Navalpakkam and L. Itti. Search goal tunes visual features optimally. *Neuron*, 53(4):605–617, 2007.

[92] A. Ngan, F. Durand, and W. Matusik. Experimental analysis of brdf models. In *Eurographics Symposium on Rendering*, pages 117–126, 2005.

[93] S. Nishida. Advancement of motion psychophysics: Review 2001–2010. *Journal of Vision*, 11(5), Dec. 2011.

[94] I. Ohzawa. Campbell-robson contrast sensitivity chart: A new rendition, 2012.

[95] Y. Pan, I. Cheng, and A. Basu. Quality metric for approximating subjective evaluation of 3-D objects. *IEEE Transactions on Multimedia*, 7(2):269 – 279, april 2005.

[96] M. G. Perkins. Data compression of stereopairs. *IEEE Trans. Commun.*, 40:684–696, 1992.

[97] R. J. Peters and L. Itti. Computational mechanisms for gaze direction in interactive visual environments. In *Proceedings of the Symposium on Eye Tracking Research & Applications*, ETRA '06, pages 27–32, New York, NY, USA, 2006. ACM.

[98] B. T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[99] M. I. Posner and Y. Cohen. *Components of Visual Orienting*, pages 531–556. 1984.

[100] G. Ramanarayanan, J. Ferwerda, B. Walter, and K. Bala. Visual equivalence: towards a new standard for image fidelity. In *ACM SIGGRAPH 2007 papers, SIGGRAPH '07*, New York, NY, USA, 2007. ACM.

[101] 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, SIGGRAPH '99*, pages 73–82, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[102] M. Reddy. Perceptually optimized 3D graphics. *IEEE Computer Graphics and Applications*, 21(5):68–75, 2001.

[103] W. Reichardt. Autocorrelation, a principle for the evaluation of sensory information by the central nervous system. In W. A. Rosenblith, editor, *Principles of Sensory Communications*, pages 303–317. John Wiley, New York, 1961.

[104] B. E. Rogowitz and H. E. Rushmeier. Are image quality metrics adequate to evaluate the quality of geometric objects. In *Human Vision and Electronic Imaging*, pages 340–348, 2001.

[105] A. M. Rohaly, J. Libert, P. Corriveau, and A. Webster. Final report from the video quality experts group on the validation of objective models of video quality assessment. Technical report, Video Quality Experts Group, VQEG, march 2000.

[106] J. Rossignac and P. Borrel. Multi resolution 3D approximations for rendering complex scenes. *Geometric Modeling in Computer Graphics*, pages 455–465, 1993.

[107] M. Roy, S. Foufou, and F. Truchetet. Mesh comparison using attribute deviation metric. *Journal of Image and Graphics*, 4:1–14, 2004.

[108] H. E. Rushmeier, B. E. Rogowitz, and C. Piatko. Perceptual issues in substituting texture for geometry. In B. E. Rogowitz and T. N. Pappas, editors, *Proceedings of SPIE Human Vision and Electronic Imaging V*, volume 3959, pages 372–383, 2000.

[109] M. V. Selst and P. Jolicoeur. A solution to the effect of sample size on outlier elimination. *The Quarterly Journal of Experimental Psychology*, pages 631–650, 1994.

[110] P. Shilane and T. Funkhouser. Distinctive regions of 3D surfaces. *ACM Transactions On Graphics*, 26(2), 2007.

[111] S. Silva, B. S. Santos, J. Madeira, and C. Ferreira. Perceived quality assessment of polygonal meshes using observer studies: A new extended protocol. In *Proceedings of SPIE Human Vision and Electronic Imaging XIII*, volume 6806, 2008.

[112] D. J. Simons and C. F. Chabris. Gorillas in our midst: Sustained inattentional blindness for dynamic events. *Perception*, 28(9), 1999.

[113] L. Stelmach, W. J. Tam, D. Meegan, and A. Vincent. Stereo image quality: Effects of mixed spatio-temporal resolution. *IEEE Transactions on Circuit and Systems for Video Technology*, pages 188–193, 2000.

[114] W. Tam, L. B. Stelmach, and P. Corriveau. Psychovisual aspects of viewing stereoscopic video sequences. In *Stereoscopic Displays and Virtual Reality Systems*, volume 3295, pages 226–235, 1998.

[115] B. W. Tatler, N. J. Wade, H. Kwan, J. M. Findlay, and B. M. Velichkovsky. Yarbus, eye movements, and vision. *iPerception*, 1(1):7–27, 2010.

[116] I. Torriente. Visual evoked potentials related to motion-onset are modulated by attention. *Vision Research*, 39(24):4122–4139, Dec. 1999.

[117] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, January 1980.

[118] K. Turkowski. Graphics gems. chapter Filters for common resampling tasks, pages 147–165. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[119] R. VanRullen. On second glance: Still no high-level pop-out effect for faces. *Vision Research*, 46(18):3017 – 3027, 2006.

[120] M. Wan, N. Zhang, H. Qu, and A. E. Kaufman. Interactive stereoscopic rendering of volumetric environments. *IEEE Trans. on Vis. and Comp. Graphics*, pages 15–28, 2004.

[121] Z. Wang, , Z. Wang, and A. C. Bovik. Why is image quality assessment so difficult? In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3313–3316, 2002.

[122] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600 –612, april 2004.

[123] Z. Wang, H. Sheikh, and A. Bovik. No-reference perceptual quality assessment of jpeg compressed images. In *Proceedings of IEEE International Conference on Image Processing, ICIP*, volume 1, pages 477–480, 2002.

[124] C. Ware. *The Environment, Optics, Resolution, and the Display*, pages 29–68. Morgan Kauffman, 2004.

[125] C. Ware. *Space Perception and the Display of Data in Space*, pages 259–294. Morgan Kauffman, 2004.

[126] 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, SIGGRAPH '01*, pages 213–220, New York, NY, USA, 2001. ACM.

[127] C. Wheatstone. Contributions to the physiology of vision. part the first. on some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical Transactions of the Royal Society of London*, 128(1):371–394, 1838.

[128] N. Williams, D. Luebke, D. J. Cohen, M. Kelley, and B. Schubert. Perceptually guided simplification of lit, textured meshes. In *Proceedings of the ACM Symposium on Interactive 3D Graphics*, pages 113–121, 2003.

[129] J.-H. Wu, S.-M. Hu, J.-G. Sun, and C.-L. Tai. An effective feature-preserving mesh simplification scheme based on face constriction. In *Proceedings of the 9th Pacific Conference on Computer Graphics and Applications, PG '01*, pages 12–21. IEEE Computer Society, 2001.

[130] H. Yamauchi, W. Saleem, S. Yoshizawa, Z. Karni, A. Belyaev, and H. P. Seidel. Towards stable and salient multi-view representation of 3D shapes. In *Proceedings of the IEEE Int. Conf. on Shape Modeling and Applications*, 2006.

[131] S. Yano, S. Ide, T. Mitsuhashi, and H. Thwaites. A study of visual fatigue and visual comfort for 3D hdtv/hdtv images. *Displays*, 23:191–201, 2002.

[132] S. Yantis and A. P. Hillstrom. Stimulus-driven attentional capture: evidence from equiluminant visual objects. *Journal of experimental psychology. Human perception and performance*, 20(1):95–107, Feb. 1994.

[133] A. L. Yarbus. *Eye movements during perception of complex objects*, pages 171–196. Plenum Press, 1967.

[134] 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.

[135] L. Zhang and T. W. J. Stereoscopic image generation based on depth images for 3D tv. *IEEE Transactions on Broadcasting*, pages 191–199, 2005.