

PERCEIVED DISPARITY REFINEMENT IN VIRTUAL ENVIRONMENTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Ufuk Çelikean
January, 2015

Perceived Disparity Refinement
in Virtual Environments
By Ufuk Çelikcan
January, 2015

We certify that we have read this thesis and that in our opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Uğur Güdükbay (Advisor)

Assoc. Prof. Dr. Tolga Çapın (Co-advisor)

Prof. Dr. Bülent Özgüç

Prof. Dr. Haşmet Gürçay

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

PERCEIVED DISPARITY REFINEMENT IN VIRTUAL ENVIRONMENTS

Ufuk Çelikcan

M.S. in Computer Engineering

Advisor: Prof. Dr. Uğur GÜDÜKBAY

Co-advisor: Assoc. Prof. Dr. Tolga ÇAPIN

January, 2015

In recent years, significant progress has been made on controlling the perceived depth range in post-production pipeline. On the other hand, different from offline production, in a virtual environment with a mobile camera, there remains a need to keep the perceived depth in the comfortable target range for the viewer. For instance, in a game environment where the stereoscopic output changes dynamically based on the user input, finding optimized stereoscopic camera parameters brings about a great challenge. Addressing such challenges of presenting a comfortable viewing setting, this work demonstrates several methods that are developed towards the goal of providing better stereo 3D experience in virtual environments.

The first part presents an approach for controlling the two stereo camera parameters, camera convergence distance and interaxial separation, in interactive 3D environments in a way that specifically addresses the interplay of binocular depth perception and saliency of scene contents. The proposed Dynamic Attention-Aware Disparity Control (DADC) method produces depth-rich stereo rendering that improves viewer comfort through joint optimization of stereo parameters. While constructing the optimization model, the importance of scene elements is considered, as well as their distance to the camera and the locus of attention on the display. The method also optimizes the depth effect of a given scene by considering the individual user's stereoscopic disparity range and comfortable viewing experience by controlling accommodation/convergence conflict. The method is validated in a formal user study that also reveals the advantages, such as superior quality and practical relevance, of considering the presented method.

In the second part, a novel method is introduced for automatically adjusting the stereo camera parameters, now also including focal length of the virtual camera lens in addition to the previous two, in a given 3D virtual scene for the scenario where there are scene elements that already have their camera parameters set for a certain perimeter and viewing angle range by the content developer and/or editor. The method, in a nutshell, computes the stereo camera parameters online by continuously scanning the scene as the virtual camera moves about it for changes in the number and the relative distribution of scene elements and the preset parameters, as well. Taking these variables into account, the method produces the camera parameters for the rest of the entire scene mainly by the employment of a radial basis function interpolation-based approach. As it works online, the framework allows for adjustment of camera parameters per scene element on-demand with an intuitively-designed interface so that the user can fine-tune the overall depth feeling of the scene.

Keywords: stereoscopic 3D, disparity control, interactive 3D, user attention, real-time graphics, accommodation/convergence conflict.

ÖZET

SANAL ORTAMLARDA ALGILANAN DİSPARİTENİN GELİŞTİRİLMESİ

Ufuk Çelikcan

Bilgisayar Mühendisliği, Yüksek Lisans

Tez Danışmanı: Prof. Dr. Uğur Güdükbay

Eş-danışman: Assoc. Prof. Dr. Tolga Çapın

Ocak, 2015

Geçtiğimiz yıllar içerisinde, algılanan derinlik alanını üretim/çekim sonrası kanallarda kontrol edebilme üzerine önemli gelişme sağlanmıştır. Öte yandan, çevrimdışı üretimden farklı olarak, hareketli kameranın bulunduğu bir sanal ortamda algılanan derinliği izleyicinin hedeflenen konfor aralığında tutmak için bir ihtiyaç mevcuttur. Örneğin, stereo görüntü çıktısının kullanıcı girdisine bağlı olarak dinamik biçimde değiştiği bir oyun ortamında, optimize stereo kamera parametrelerini bulmak hayli zorlu bir uğraştır. Konforlu bir seyir sağlamanın bu tür zorluklarını ele alarak, bu eserde sanal ortamlarda daha iyi üç boyutlu stereo görüntü tecrübesi sunmaya yönelik metotlar ortaya konmuştur.

İlk kısımda, üç boyutlu etkileşimli ortamlarda binoküler derinlik algısının ve sahne içeriğinin karşılıklı etkilerini göz önünde bulundurarak stereo kameraların iki ana parametresini oluşturan kamera-aksları arası uzaklığı ve kameraların yakınsama uzaklığını kontrol etmek üzere bir yaklaşım ortaya konulmuştur. Bu önerilen İlgi-Bilinçli Dinamik Disparite Kontrolü metodu stereo parametrelerinin bütünleşik optimizasyonunu sağlamak suretiyle zengin derinlikli stereo görsel gerçekleştirme üreterek izleyicinin seyir konforunu iyileştirmektedir. Optimizasyon modeli geliştirilirken sahne elemanlarının tek tek önemleri ve aynı zamanda her birinin kameraya ve sahnedeki ilgi odağına uzaklığı dikkate alınır. Önerilen metot kullanıcının bireysel stereo disparite alanını ve konforlu seyir tecrübesini göz önünde bulundurarak akomodasyon-yakınsama ikiliğini kontrol altında tutarak sahnenin derinlik efektini de optimize etmektedir. Yapılan kurallı kullanıcı deneyleri, metodun işlerliğini ortaya koymuş ve aynı zamanda yüksek nitelik ve pratikte uygunluğunu ortaya çıkarmıştır.

İkinci kısımda, daha önceki iki stereo parametresinin ve bunlara ilaveten lens odak uzunluğunun içerik oluşturucusu veya editörü tarafından sahnedeki belli başlı elemanlara belirli bir çevre ve görüş açısı aralığı için halihazırda belirlenmiş olduğu bir senaryoda kameranın stereo parametreleri otomatik olarak ayarlamaya yönelik bir metot sunulmuştur. Metot, kısaca, kamera sahne içerisinde hareket ederken sahne elemanlarının sayısındaki ve dağılımındaki değişimler ve elemanların atanmış parametreleri için sürekli sahneyi taramak vasıtasıyla stereo kamera parametrelerini gerçek zamanlı olarak hesaplamaktadır. Bu değişkenleri hesaba katarak, ana olarak radyal taban fonksiyonları ile interpolasyon temelli bir biçimde sahnenin girekalanı için uygun stereo kamera parametrelerini üretmektedir. Ortaya konan sistem, gerçek zamanlı çalıştığı için, sezgisel tasarım çerçevesinde gerçekleştirilen kullanıcı ara yüzü vasıtasıyla kullanıcı her arzu ettiğinde her bir sahne elemanı için atanmış parametrelerin değiştirilebilmesini sağlayarak kullanıcının sahnenin genel derinlik hissini kişisel olarak şekillendirebilmesine olanak verir.

Anahtar sözcükler: stereo 3D, disparite kontrol, etkileimli 3D, kullanıcı ilgisi, gerek-zamanlı grafikler, akomodasyon-yakınsama ikilii.

Acknowledgement

First of all, I am deeply grateful to Assoc. Prof. Dr. Tolga apın who has been my advisor since the very beginning of this work. I feel very lucky to have found the chance to work with him. His guidance has always been invaluable as well as his motivation.

I am thankful to Professor Uğur G d kbay who has also been my advisor this past half-year and been very helpful in all aspects of this work.

Professors B lent  zg   and Hařmet G r ay, I thank for participating in my thesis committee, for accepting to read and evaluate this work, and for their valuable comments.

I want to thank Dr. Tolga apın, Prof. Uğur G d kbay and Prof. B lent  zg   once again together for being an excellent team of mentors on my studies, projects and research at Bilkent University. I learnt a lot from all of them.

I also want to thank Professor Hařmet G r ay once again for introducing me to new and exciting ideas and opportunities that eventually became the milestones of my career.

I feel a deep gratitude to my PhD advisor Professor Ertem Tuncel at University of California, Riverside. I thank him a lot for his encouragement and support when I engaged upon a significant career change right in the middle of my study and for letting me continue my work and finish writing my PhD dissertation in Turkey.

I would like to thank Aytek Aman and Ateř Akaydın for kindly supplying some of the 3D human mesh models that were used in preparation of the virtual scenes; and Sami Arpa and the 3dios Productions Company for providing the 3D display equipment that was used for testing and the user-studies.

I would also like to mention the financial support of TUBITAK (Scientific and

Technical Research Council of Turkey) since this research began as a part of the Perceptually-Based 3D Graphics project which was supported by TUBITAK 1001 program (grant number 110E029).

I can never thank my family enough for their infinite compassion and total support during all my studies, and, as a matter of fact, all my life. I am thankful to my granny who never leaves my well-being out of her prayers. And I am eternally thankful to my mother who has always supported me and my decisions no matter what; and always encouraged me to follow my dreams no matter where they may take me.

Above and beyond all, I have the utmost gratitude to my wife Merve. She has been my anchor; she has been my compass; she has been the light that shone on this work as she has always been the light of my life ever since the moment we first held hands.

To my family,

my mother,

and

Merve

Contents

I Preliminary on Stereo Vision and Disparity Refinement in Virtual Environments	1
1 Introduction	2
2 Background	4
2.1 Depth Perception.	4
2.2 Stereo Geometry.	5
2.3 Accommodation/Convergence Conflict.	7
3 Related Work	9
 II Attention-Aware Disparity Control in Interactive Environments	 11
4 Introduction	12
5 Approach	14
5.1 Depth Range Control (DRC)	16

5.2	Dynamic Attention-Aware Disparity Control (DADC)	17
5.2.1	Depth Range Calculation.	18
5.2.2	Analysis of Scene Contents.	18
5.2.3	Optimization of Stereo Parameters with Active Depth Control.	20
6	Experimental Evaluation	24
6.1	Subjects	24
6.2	Equipment	25
6.3	Scenes	25
6.4	Procedure	25
6.5	Assessment of Contents	26
7	Results	29
8	Conclusion	34
III	RBF Interpolation-Based Disparity Refinement	35
9	Radial Basis Function Interpolation	36
9.1	Scattered Data Interpolation	37
9.2	Introduction to Radial Basis Functions	39
9.2.1	Basis Functions	41

10 Approach	45
10.1 Content-Preparation	45
10.2 Main Phase	47
10.2.1 Edit Mode	47
10.2.2 Play Mode	51
10.3 System	53
10.4 Testing	53
11 Conclusion	56
Bibliography	57

List of Figures

2.1	A virtual camera setup with parallel sensor-shift (left) and the corresponding reconstruction of stereoscopic 3D scene	5
4.1	(a) An example capture of the scene with Naive method (b) Disparity limit calibration (c) Depth map of a captured scene (d) Significance score coloring of scene elements (e) Output stereoscopic image with DADC (f) Capture of the scene with DADC	13
5.1	Overview of the main phase of our approach: (a) In the first stage, visible scene depth extrema information is gathered. This information in combination with the data collected from the disparity calibration phase is fed into the optimization as system constraints. (b) The scene content analysis stage, as outlined in Algorithm 1, extracts $\{S, Z, R\}$ information of significant elements in the visible scene. (c) The system searches for the optimal parameter set $\{Z_c, t_c\}$ seeking to keep significant scene elements inside the comfort zone while maximizing the perceived depth feeling. The system output is finalized by applying <i>temporal control</i> to the optimization output.	15
6.1	First row shows snapshots of outdoor scene, second row shows of indoor scene.	26

6.2	Presentation of test material	27
7.1	Charts describing the subjects' ratings and averages based on 5-point Likert scale for our method and the compared methods. In each chart, the average grade is indicated in a circle.	30
7.2	Aggregated results from our session comparison questionnaires demonstrating relative user preferences of our DADC method in percentages. Scores are relative to Naive method in the first row and DRC method in the second.	31
7.3	Depth charts of an evaluated scene for the first hundred frames with (a) Naive method (b) DRC (c) DADC	33
9.1	Sample data from test function.	37
9.2	2D view (left) and 3D view (right) of the data set Hurrungane (23092 points).	38
9.3	Conceptual representation of RBF interpolation (a) Real terrain surface (b) Surface with sampled location indicated (c) Cross-section of the RBF interpolated surface.	39
9.4	Comparison of (a) inverse distance weighted interpolation (b) radial basis function interpolation.	40
9.5	Gaussian RBFs plotted with (a) $r_0 = 1/3$ (b) $r_0 = 1$ (c) $r_0 = 2.5$, on the same domain.	43
10.1	Overview of the approach is presented. The framework for the approach necessitates a specific type of content-preparation phase as a preliminary. Once the content is ready to be processed by the framework, the main phase takes over. There are two alternating parts to the main phase. These are named as the Play Mode and the Edit Mode of the framework.	46

10.2 Sample screen-captures of the user interface from an Edit Mode user session.	50
10.3 Diagram of a sample path used in non-interactive testing of the approach.	54
10.4 Available stereo-output settings of the framework (a) red-cyan anaglyph output (b) side-by-side output (before being overlapped by the display device).	55

List of Tables

2.1	The review of the perceptual effects of stereo parameters (adapted from Milgram and Kruger [6])	6
-----	---	---

List of Algorithms

1	Scene content analysis algorithm	19
2	Edit Mode Algorithm	48
3	Play Mode Algorithm	52

Parts of this thesis including Figures 2.1, 4.1, 5.1, 6.1, 6.2, 7.1, 7.2, 7.3 and Table 2.1, in addition to the text within Parts I and II, are reprinted from

Ufuk Celikcan et al., 2013: *Attention-aware disparity control in interactive environments*. The Visual Computer, 29.6-8, 685-694, Copyright © 2013, Springer.

with kind permission from Springer Science and Business Media.

Part I

Preliminary on Stereo Vision and Disparity Refinement in Virtual Environments

Chapter 1

Introduction

Recent advances in stereoscopic displays and 3D TVs, 3D digital cinema, and 3D enabled applications have increased the importance of stereoscopic content creation and processing. However, several challenges remain in providing realistic but comfortable viewing experience to users with stereoscopic products. One of the principal challenges is a need for applying the underlying principle of 3D perception of the human visual system and its capabilities/limitations for displaying content in stereoscopic displays.

Binocular viewing of a scene is created from two slightly different images of the scene in the two eyes. These views are produced by stereoscopic rendering parameters, which are camera separation and convergence distance of cameras. The difference in the views, or screen disparities, create a perceived depth around the display screen. The main concern of stereoscopic 3D content creation is determining the comfortable range of this perceived depth, also called *the comfort zone*.

Recent research has made progress in controlling the perceived depth range, mostly in post production pipeline [1–3]. On the other hand, different from offline production, in an interactive environment where the position of the camera is dynamically changing based on the user input, there is a need for a control system to keep the perceived depth in the comfortable target range. Examples

for such controllers are the work of Lang et al. [3] for post-production disparity range adjustment and the work of Oskam et al. [4] for real-time disparity range adaptation.

An example for an interactive setting is a game environment where the stereoscopic output changes dynamically. For such an environment, finding optimized stereoscopic camera parameters, i.e., camera convergence distance and interaxial separation to retarget dynamic scene depth to comfortable target depth range brings a great challenge. Even though previous works manage to control and limit the perceived depth to comfort zone of the users, there is also a need to define parameters for preventing the violation of accommodation/convergence conflict. This conflict can cause severe consequences in such interactive stereoscopic environments in long-term use. The inability of fusion, also called diplopia, is one of the major problems that emerge because of accommodation/convergence conflict, and further problems include eye-strain, visual fatigue and even headache after prolonged exposure.

Chapter 2

Background

In this chapter we summarize the basic principles and the characteristics of binocular vision and stereo geometry.

2.1 Depth Perception.

Depth cues, which help the human visual system to perceive spatial relationships between objects, constitute the core part of depth perception. These visual cues can be categorized as pictorial, oculomotor, binocular, and motion-related cues [5]. Pictorial cues, such as occlusion, shadow, shading, relative size, relative height, texture gradient, are extracted from a single and flat 2D view; whereas oculomotor depth cues represent depth perception that is obtained through eye movements. Motion parallax, motion perspective, and kinetic depth are the motion based depth cues. The two types of binocular depth cues are named as convergence and retinal disparity, which are covered in detail in the following.

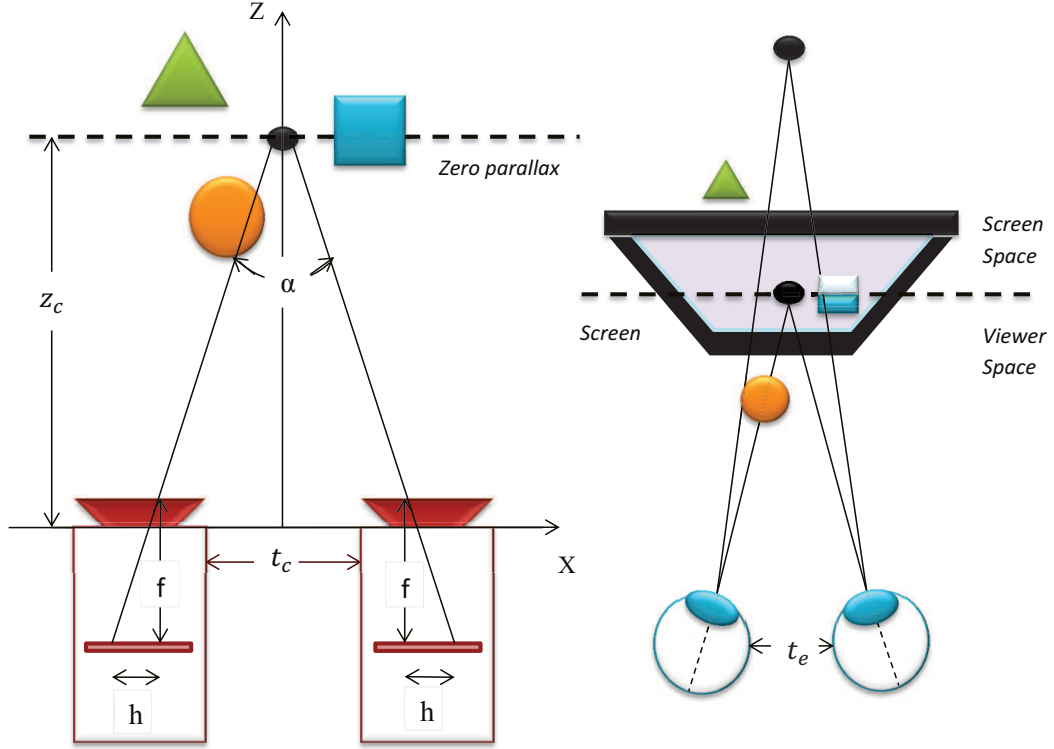


Figure 2.1: A virtual camera setup with parallel sensor-shift (left) and the corresponding reconstruction of stereoscopic 3D scene

2.2 Stereo Geometry.

The binocular depth cue makes use of the fact that left and right eyes view the world from slightly different angles, which results in slightly different retinal images, forming binocular vision. The parameters that are used in the human visual system by their real world correspondences are *binocular disparity* and *vergence*. Binocular disparity represents the difference between the two eyes; whereas vergence arises due to eye movements and allow fixating at a point of interest.

In stereoscopic image creation, the main difficulty arises while controlling the stereoscopic camera parameters. There are two principal parameters for disparity: *interaxial separation* (t_c) and *convergence distance* (Z_c), as illustrated in Figure 2.1. While convergence distance corresponds to the distance between the camera and the plane in focus, the interaxial separation corresponds to the separation

between the two cameras. The camera separation, or interaxial separation (t_c) directly affects the disparity and eventually the amount of depth perceived in the final image. The convergence distance, on the other hand, does not affect the overall perceived depth, but increasing the convergence distance decreases the screen parallax. Table 2.1 summarizes the perceptual effects of the stereoscopic camera parameters.

Table 2.1: The review of the perceptual effects of stereo parameters (adapted from Milgram and Kruger [6])

		Disparity	Perceived Depth	Object Size
t_c	Increases	<i>Increases</i>	<i>Increases</i>	<i>Constant</i>
	Decreases	<i>Decreases</i>	<i>Decreases</i>	<i>Constant</i>
Z_c	Increases	<i>Decreases</i>	<i>Shifts Forward</i>	<i>Constant</i>
	Decreases	<i>Increases</i>	<i>Shifts Backward</i>	<i>Constant</i>

Given the parallel camera geometry in Figure 2.1, the image disparity of an object with scene distance Z depends on interaxial separation (t_c) and convergence distance (Z_c), and is given as:

$$d = ft_c \left(\frac{1}{Z_c} - \frac{1}{Z} \right) \quad (2.1)$$

In this equation, f denotes the focal length of the cameras. The conversion from image disparity d to screen parallax p simply requires scaling the image disparity from image sensor metric to display size metric, by multiplying it with a scale factor W_s/W_i , where W_i and W_s denote the image sensor width and screen width respectively.

$$p = d(W_s/W_i) \quad (2.2)$$

While maintaining stereoscopic depth, the viewer reconstructs a point for each object on and around the screen. The reconstructed depth Z_r of this point, while the viewer is observing from a physical distance Z_w , is given as:

$$Z_r = \frac{Z_w t_e}{t_e - p} = \frac{Z_w t_e}{t_e - d(W_s/W_i)} \quad (2.3)$$

where t_e is the human interocular distance, for which the physiological average is approximately 65 mm.

The convergence distance gives the distance where the two cameras converge; and on the plane at that distance the retinal positions of objects appear at the same point which results in objects appearing at the physical screen surface ($Z = Z_c$). This condition is called *zero parallax setting*. Two conditions occur when object distances Z are different from Z_c . In the first case, ($Z > Z_c$), the object appears inside the screen space, which is viewed behind the display screen. When this condition occurs, the object has a positive disparity, or screen parallax. On the other hand, in the case ($Z < Z_c$), the object has a negative disparity, or parallax. These objects appear as if they are physically located in front of the screen.

Physiological experiments have proven that the human visual system has more tolerance to positive parallax than negative parallax [7]. However, it is still restricted to comfortably perceive all objects which appear in positive or negative parallax regions. It has been shown that locating the scene in a limited area around the screen surface gives more reasonable results for avoiding accommodation/convergence conflicts.

2.3 Accommodation/Convergence Conflict.

The conclusion pointed out by several earlier studies [8] on the issue of stereoscopic comfort zone is that the amount of perceived depth in stereoscopic displays should be limited; and the conflicts related to accommodation and convergence should be controlled. The accommodation/convergence conflict happens for all

planostereoscopic displays, i.e. displays where the views are presented on a planar screen. This conflict is caused by the fact that when looking at the stereoscopic 3D display, viewer's eyes converge on the reconstructed depth Z_r , while they are forced to focus on the display plane. This is in contrast to natural vision in the real world, where the human visual system operates such that the eyes converge and accommodate at the same point.

Chapter 3

Related Work

With the recent advances in stereoscopic systems, the focus on stereoscopic camera control has gained momentum and a number of techniques have been proposed for stereoscopic post-production pipeline and editing of stereoscopic images.

3D Camera Systems and Stereo Acquisition. The conventional way for capturing real scenes is with two physical camera equipments. One of the recent approaches which focus on production of high quality stereoscopic content capture is presented by Zilly et al. [9]. This system analyzes the captured scene by two real cameras and specifies the proper camera calibration parameters. Heinzle et al. [10] focus on controlling the rig directly, with a control loop that consists of capture and analysis of 3D stereoscopic parameters.

Stereoscopic editing on still images. Recent work on stereoscopic image editing focuses on correction of imperfect stereoscopic images and videos. Koppal et al. [11] present an editor for live stereoscopic shots. They concentrate on the viewer's experience and propose modifying camera parameters in the post processing as well as previewing steps. Lang et al. [3] present a nonlinear disparity mapping method in order to retarget the depth range in the produced stereoscopic images and videos to different displays and viewing conditions. Didyk et al. [12] have also recently proposed a disparity model that estimates the perceived

disparity change in processed stereoscopic images, and perform psychophysical experiments in order to derive a metric for modeling disparity. Didyk et al. [2] also proposed an extended luminance-contrast aware disparity model, and presented disparity retargeting as one of its applications.

Stereo parameter adjustment in virtual environments. Post processing and image shifting methods are used for retargeting disparity in offline applications such as digital cinema and 3D content retargeting. On the other hand, interactive applications require real-time techniques. Among recent works, the geometrical framework to map a specified depth range to the perceived depth range is described by Jones et al. [13]. Their method is proposed for generating still images, but it can also be used for virtual scenes. Oskam et al. [4] present a controller for finding camera convergence and interaxial separation, which gives a final disparity value for the viewed frame. These parameters change automatically by taking minimum and maximum scene depth values into account in order to handle excessive binocular disparities which are generated because of unpredictable viewer motion.

Part II

Attention-Aware Disparity Control in Interactive Environments

Chapter 4

Introduction

In this first part of our work [14], as outlined in Figure 4.1, we aim to address the challenges of presenting a comfortable viewing experience to users in an interactive scene, by controlling and limiting target depth range to the comfort zone and eliminating accommodation/convergence violations as much as possible. For mapping scene depth to the specific depth range, our method automatically finds optimized stereo camera parameters in real-time. In order to avoid accommodation/convergence conflict, we consider the distribution and importance of scene elements. For this purpose, the convergence plane is moved so that significant elements are shown with relatively sharper focus. This motivation comes from that the location of the convergence plane, on which scene elements are captured with exactly zero disparity, should tend to be nearer to elements with higher significance during the search, assuming each element of interest in the scene content carries a significance score that is assigned by the content creator.

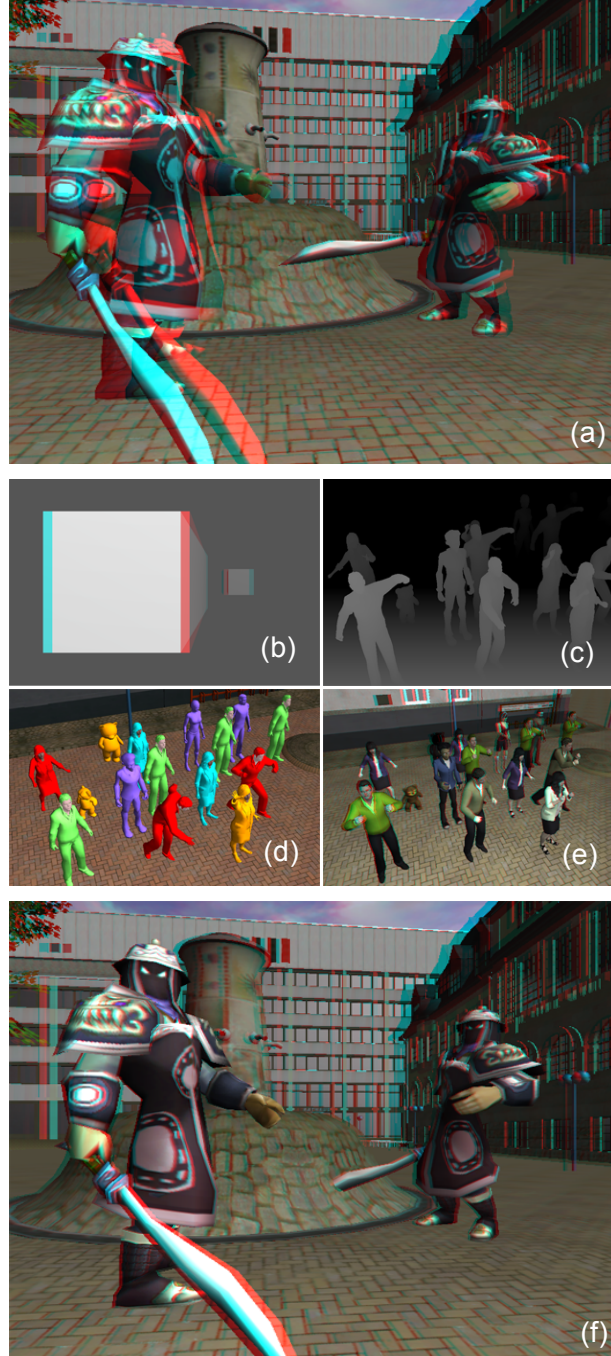


Figure 4.1: (a) An example capture of the scene with Naive method (b) Disparity limit calibration (c) Depth map of a captured scene (d) Significance score coloring of scene elements (e) Output stereoscopic image with DADC (f) Capture of the scene with DADC

Chapter 5

Approach

Our approach (Figure 5.1) consists of a calibration phase and a main phase. In the calibration phase, the depth perception range of the user is obtained interactively. Perceived depth range is changeable in light of user’s personal stereoscopic comfort limits. For this purpose, the user designates the personal disparity extrema, so that the disparity is not too high in order to avoid eye-straining visual artifacts like diplopia, or too low resulting in low depth feeling. This calibration stage is needed to be performed only once per user, before starting the interactive stage.

During the main phase, for the incoming frame, we first analyze the depth range of the scene from the given view position. Consecutively, we perform an analysis of the scene contents, in terms of their layout under the given viewing condition. For this purpose, for each object in the view, we consider its significance score, its distance to the camera and center of display, and construct an optimization problem that we solve to calculate the stereo parameters, t_c and Z_c . Our method also makes use of temporal coherency constraint, so that the stereo parameters change smoothly between frames.

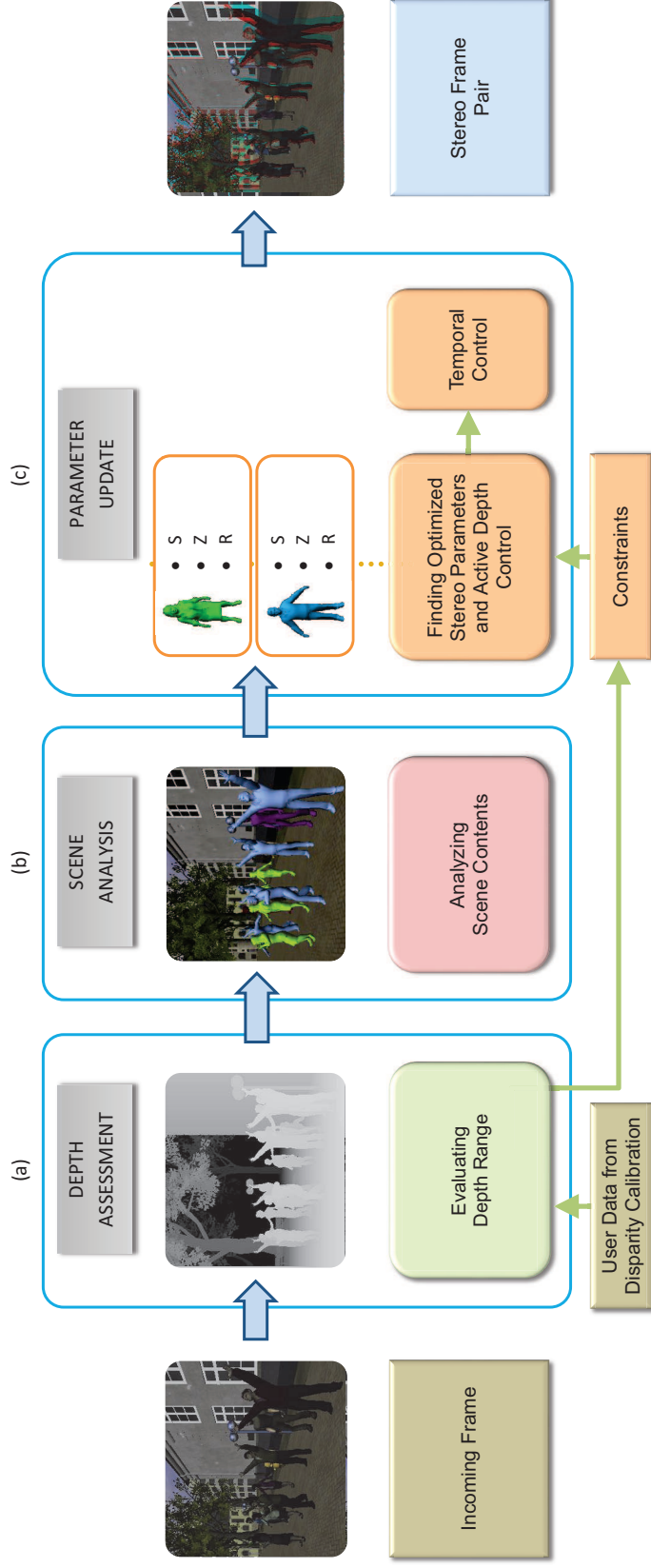


Figure 5.1: Overview of the main phase of our approach: (a) In the first stage, visible scene depth extrema information is gathered. This information in combination with the data collected from the disparity calibration phase is fed into the optimization as system constraints. (b) The scene content analysis stage, as outlined in Algorithm 1, extracts $\{S, Z, R\}$ information of significant elements in the visible scene. (c) The system searches for the optimal parameter set $\{Z_c, t_c\}$ seeking to keep significant scene elements inside the comfort zone while maximizing the perceived depth feeling. The system output is finalized by applying *temporal control* to the optimization output.

5.1 Depth Range Control (DRC)

Our method is an extension of the methods that control the depth range in a given scene. Among which, the most widely used one is Depth Range Control (DRC) method and our approach includes this method as a special case. Therefore, we first explain DRC, before discussing our approach in detail.

It is possible to approximate the perceived disparity by geometrically modeling the stereoscopic vision with respect to a given depth-range which may be adjusted by the viewer. According to this approach, interaxial separation and convergence distance can be formulated [8] by using similar triangles in the stereo vision geometry. This, for an image-shift camera convergence setup, results in:

$$Z_c = \frac{Z_{max}Z_{min}(d_{max} - d_{min})}{(Z_{max}d_{max} - Z_{min}d_{min})} \quad (5.1)$$

$$t_c = \frac{Z_{max}Z_{min}(d_{max} - d_{min})}{f(Z_{max} - Z_{min})} \quad (5.2)$$

where

- Z_{max} : The distance between the camera and the farthest object in the virtual world.
- Z_{min} : The distance between the camera and the nearest object in the virtual world.
- d_{max} : Maximum disparity, i.e., the positive disparity of the farthest object.
- d_{min} : Minimum disparity, i.e., the negative disparity of the nearest object.

Jones et al. [13] applied this model to adjust the target depth range of still images only. Guttman et al. [15] used the model for recreating stereographic sequences

from 2D input by estimating the correct target depth distribution and optimizing the target disparity map. Oskam et al. [4] developed a similar method for interactive applications for optimizing stereo rendering parameters with respect to control points each assigned a certain desired depth. In the special case with only two constraints, one for each depth extremum, their system simplifies to Eq. 5.1 and Eq. 5.2 above.

In any case, the mentioned methods are based on mapping the depth range, without consideration of the distribution of the objects in the scene. Therefore, we believe that employing DRC method alone is not sufficient in enhancing the perceived stereo vision effect, as psychological elements directly affect the creation of stereo vision, especially in interactive applications. In this regard, we develop an attention-aware system which involves real-time analysis of scene contents as well as depth range assessment for user-specific disparity control.

5.2 Dynamic Attention-Aware Disparity Control (DADC)

As overviewed in the previous section, it is known that objects which are located in the 3D comfort zone of the user are easier to be observed. Thus, significant scene elements that draw user’s attention should be located closer to this region. However, in a pre-produced interactive scene, it is necessary to move the convergence plane instead, placing it as near as possible to the region that attracts the user’s attention the most, while maintaining the total disparity of the scene as high as possible and not violating the user’s disparity range.

With this goal in mind, the main phase of our stereoscopic 3D control system is composed of the following three consecutive stages.

5.2.1 Depth Range Calculation.

Since the maximum and the minimum distances observed by the virtual camera have a direct effect on screen disparity and thus the depth experienced by the user, we need to gather visible scene depth extrema information. This is achieved by a number of min-max reduction passes on the depth buffer [16]. The system runs this normally costly procedure in real-time (i.e., within the allowed per-frame time budget) by efficient utilization of the GPU.

This information in combination with the data collected from disparity calibration of the user is fed into the optimization as system constraints, and is also used in the two special non-optimization cases, as explained in detail later.

5.2.2 Analysis of Scene Contents.

Having adopted interactive environments as our main consideration, we make the following arguments in conjunction with our objective function that is explained in the next section:

- The user navigates towards scene elements that attract his attention more.
- The user tends to have significant scene elements centered in his view

Based on these assumptions, we evaluate the overall significance of a scene element with respect to the three criteria below:

- S : significance score of the element.
- Z : forward distance of the element from camera.
- R : radial distance of the element from forward camera axis.

Algorithm 1 Scene content analysis algorithm

```
1: procedure SCENEANALYSIS
2:
3:    $e[] \leftarrow \text{getSignificantElements}()$ 
4:    $\triangleright$  Acquiring all significance score assigned elements in the current scene
5:
6:    $j \leftarrow 0$ 
7:   for  $\forall e[i]$  do
8:     if  $e[i]$  is visible in the current frame then
9:
10:       $e[i].Z \leftarrow \text{ForwardDistanceFromCamera}()$ 
11:
12:      if  $e[i].Z \leq D_{max}$  then
13:         $\triangleright D_{max}$ : maximum forward distance allowed
14:
15:         $o[j] \leftarrow e[i]$   $\triangleright$  implies  $o[j].S \leftarrow e[i].S$  and  $o[j].Z \leftarrow e[i].Z$ 
16:
17:         $o[j].R \leftarrow \text{RadialDistanceFromCameraAxis}()$ 
18:         $j \leftarrow j + 1$ 
19:
20:      end if
21:
22:    end if
23:  end for
24:
25:  return  $o[]$ 
26:
27: end procedure
```

Here, we assume that scene elements had been assigned significance scores by the content creator that would appropriately predict the user's relative attention towards them such that e.g., in a first-person game environment the autonomous enemies should have been assigned higher scores compared to other scene elements.

Our scene content analysis algorithm progresses as outlined in Algorithm 1.

5.2.3 Optimization of Stereo Parameters with Active Depth Control.

For establishing our objective function to be optimized, we first formulate an energy term $E_o(Z_c, t_c)$ that penalizes the distance of the convergence plane from scene elements with relatively higher significance score and/or with relatively lower radial distance from the user’s center of attention.

In order to minimize visual artifacts like ghosting associated with significant scene elements, the higher the significance score of an element the closer convergence plane should move towards it through minimization of $E_o(Z_c)$ thus keeping that element in relatively sharper focus.

Several methods have been proposed for computational modeling of visual attention [17]. Studies have converged on a two-component framework for attention; where viewers selectively direct their attention in an image, to objects in a scene using both (i) bottom-up, image-based saliency cues and (ii) top-down, task-dependent cues.

For precise detection of the center of attention, a perceptually based system should include some sort of eye-tracking technology as it deals with the extent of features across the user’s retina or at least head-tracking technology that mimics eye-tracking by the observation that resting eye gaze can approximately track head orientation. However, when no eye or head tracking exists, as is the case with most stereoscopic viewing settings, we are to conform to the assumption [18] that the user always looks toward the center of the display device. Considering this, by minimizing $E_o(Z_c)$, the resulting convergence plane should also move closer towards scene elements with relatively less radial distance from the forward axis of virtual camera i.e., display center.

Following this line of thought, $E_o(Z_c)$ is formulated as

$$E_o(Z_c) = \sum_{i=1}^n \frac{S_i}{R_i^2} (Z_i - Z_c)^2, \quad (5.3)$$

where n is the number of significant scene elements found in the scene analysis stage.

We use a second energy term $E_d(Z_c, t_c)$ which pursues to maximize total scene disparity and, therefore, total perceived depth. Formulation of $E_d(Z_c, t_c)$ follows the regular disparity calculation (Eq. 2.1) s.t.

$$E_d(Z_c, t_c) = \sum_{i=1}^n S_i f t_c \left(\frac{1}{Z_c} - \frac{1}{Z_i} \right), \quad (5.4)$$

hence aggregating weighted disparity associated with each significance assigned scene element. Here, disparities are also weighted with respective significance scores S_i .

We construct the objective function as the total energy function $E(Z_c, t_c)$ s.t.

$$E(Z_c, t_c) = \hat{E}_o(Z_c) - \hat{E}_d(Z_c, t_c), \quad (5.5)$$

Here $\hat{E}_o(Z_c)$ and $\hat{E}_d(Z_c, t_c)$ are the normalized energies s.t.

$$\hat{E}_o(Z_c) = E_o(Z_c) / (Z_{max} - Z_{min})^2, \quad (5.6)$$

$$\hat{E}_d(Z_c, t_c) = E_d(Z_c, t_c) / (d_{max} - d_{min}), \quad (5.7)$$

This way with appropriate normalization, the need to express $E(Z_c, t_c)$ as a weighted sum of $E_o(Z_c)$ and $E_d(Z_c, t_c)$ with weights that are to be fine-tuned for every different setting and every different user is avoided.

Consequently, by minimizing $E(Z_c, t_c)$, the system searches for the optimal parameter set by mediating the minimization of $E_o(Z_c)$ with the maximization of $E_d(Z_c, t_c)$, thus seeking to keep significant scene elements inside the comfort zone while maximizing the perceived depth feeling.

The system minimizes $E(Z_c, t_c)$ subject to constraints:

$$d_{max} \geq ft_c \left(\frac{1}{Z_c} - \frac{1}{Z_i} \right) \geq d_{min}, \quad \forall i | 1 \leq i \leq n, \quad (5.8)$$

with d_{max} and d_{min} obtained from disparity calibration phase. The constraints ensure that during the optimization scene depth is actively mapped into the perceivable depth range of the user as initially determined.

The nonlinear system is globally optimized within the parameter space by improved stochastic ranking based evolutionary strategy (ISRES) algorithm [19]. The ISRES algorithm, a major representative of the state of the art in constrained optimization, is based on a simple evolution strategy augmented with a stochastic ranking that decides by carrying out a comparison, which utilizes either the function value or the constraint violation. With the incorporation of ISRES implementation in NLOpt library [20] using modern multi-core processor technology via multi-threading, we achieve optimization at interactive speed so that the system is able to produce the updated stereo parameters continually as e.g., the user navigates through a scene.

5.2.3.1 Frames with only a single element of interest

When the system finds a single significance assigned element visible, it places the element at the screen i.e., $Z = Z_c$ and computes interaxial separation using the

DRC method.

5.2.3.2 Frames without an element of interest

For frames containing no significance assigned element, our system switches to complete DRC mode and computes the stereo parameters accordingly.

5.2.3.3 Temporal Control

Stereoscopic 3D rendering parameters are recalculated for each frame as a desired solution. On the other hand this situation may cause undesired visual artifacts if changes in parameters occurring between consecutive frames are considerably high or happening more frequently than tolerable. In order to uphold temporal coherence, the system produces the final parameter set for the processed frame by passing each newly computed parameter through a threshold function $f(\cdot)$ s.t.

$$f(x(t)) = \begin{cases} x(t-1) + x_1, & \text{if } x(t) - x(t-1) \leq x_1; \\ x(t-1) + x_2, & \text{if } x(t) - x(t-1) \geq x_2; \\ x(t-1) + k(x(t) - x(t-1)), & \text{otherwise.} \end{cases} \quad (5.9)$$

where $x_1 \in \mathbb{R}^-$, $x_2 \in \mathbb{R}^+$ and k is chosen to be $0 < k < 1$.

Chapter 6

Experimental Evaluation

To evaluate our method, we tested it in two different scenes in pair-wise comparisons to the DRC only approach and the Naive approach. The Naive approach uses fixed stereo parameters that are initialized with DRC method at the beginning of each test session.

6.1 Subjects

We recruited 15 subjects, with a mean age of 25. The subjects were among voluntary undergraduate and graduate students with computer science background; and most of them did not have previous detailed experience on rendering on stereoscopic displays. Prior to the study, each subject candidate was tested for proper stereoscopic visual acuity using random dot stereogram test and those who failed the test did not participate in the user study. The subjects were not informed about the purpose of the experiment.

6.2 Equipment

We used a 2.20 GHz Quad-Core laptop with 6 GB RAM for rendering; and a 40 inch 3D display with active shutter glasses, with a resolution of 1920 x 1080. The subjects were seated at a viewing distance of 2m.

6.3 Scenes

We built two interactive scenes (Figure 6.1) for the tests. The first scene contains an indoor setting, where several groups of human characters, each of which performing various gestural movements, randomly distributed in a room. The second one contains an urban outdoor setting that presents a more dynamic environment in terms of variety of characters and their actions, as well. Virtual characters were assigned relatively higher significance in both scenes. In each test, the user was asked to navigate freely in the environment.

6.4 Procedure

Subjects were given written instructions describing the task that needed to be performed, and the attributes that need to be rated.

Our user study procedure was consistent with the ITU-R BT.2021 Recommendation, on subjective methods for the assessment of stereoscopic 3D systems [21]. For the experiment design, we have followed the double stimulus continuous quality scale (DSCQS) method. According to this procedure, subjects are shown a content, either test or reference; after a brief break, they are shown the other content. Then, both contents are shown for the second time, to obtain the subjective evaluations. This process is illustrated in Figure 6.2.

To evaluate our method vis-a-vis the two other methods (DRC and Naive), we performed the tests in pairs of sessions for each subject. For each pair of sessions,

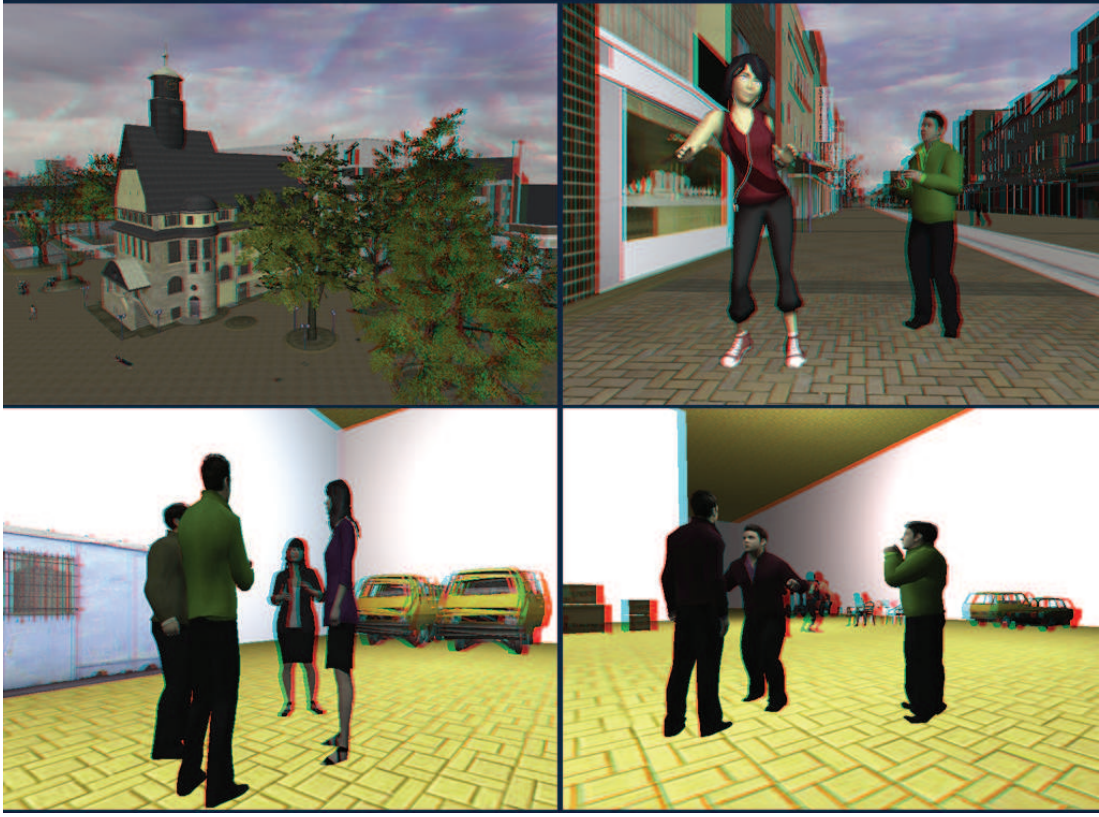


Figure 6.1: First row shows snapshots of outdoor scene, second row shows of indoor scene.

our method is used in the test content session while the compared method, either Naive or DRC, is used in the reference content session. The order of the reference and the test sessions in a pair and the order of the compared methods in consecutive pairs were both determined randomly. The subjects were not informed about either order. This set of tests were executed for each of our interactive scenes. Between the two sets of tests, a two minute break was introduced to relax eye muscles. Overall, eight test sessions were evaluated by each subject.

6.5 Assessment of Contents

Subjects evaluated both test and reference content sessions of all cases separately, with respect to three criteria: quality, depth, and comfort. These three criteria

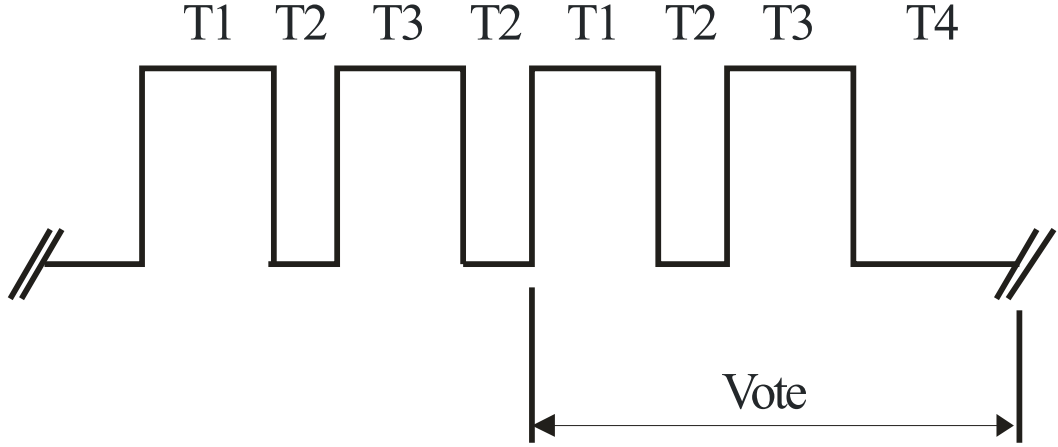


Figure 6.2: Presentation of test material

are commonly used in the perceptual evaluation of stereoscopic contents [21]. The meaning of each criterion was explained to the subjects before the experiments. The motivation behind selecting these grading criteria is as follows:

- *Image Quality*: Image quality denotes the perceived overall visual quality of the shown content. Ghosting, defined as the incomplete fusion of the left and right image so that the image looks like a double exposure, is a critical factor determining the image quality of a stereoscopic content. A good quality 3D stereo image should eliminate the ghosting effect.
- *Perceived Depth*: This criterion measures the apparent depth as reported by the user, so that the effect of the methods on apparent depth should be taken into account.
- *Visual (Dis)comfort*: refers to the subjective sensation of discomfort that can be associated with improperly set stereoscopic parameters by the different algorithms. A good quality 3D stereo image should provide a comfortable viewing experience.

For assessment of the content, we also followed a methodology following the ITU-R BT.2021 Recommendation. We first asked the subjects to rate the quality, depth, and comfort of both the reference and test sessions separately, by filling

out a 5-point Likert scale for each session. For assessment of quality, depth, and comfort, we used the discrete scale with the labels “bad”, “poor”, “fair”, “good”, and “excellent”. Then, at the end of each session pair, we also asked the subjects to compare between the two sessions. For this purpose, we asked the following questions in the evaluation form:

- Which session provided better image quality?
- Which session offered more depth?
- Which session was more comfortable to watch?
- Which session provided better overall quality?

Chapter 7

Results

In order to analyze the user assessments, we computed the average scores for user ratings, as well as user preferences. Figure 7.1 illustrates the rating results for image quality, depth and comfort measures. The results show that our method yields better average than other approaches in all measures. Our DADC method achieved a considerable improvement particularly in the stereoscopic image quality, due to the fact that our method ensures the elimination of ghosting effect of the elements of interest in the scene to a significant extent. Regarding the assessment of image depth, the average rating of our method is slightly better than the other two methods, but less number of subjects have evaluated the depth impression of our method as “bad” or “poor”, compared to the other methods. The comfort ratings also reveal that our method is generally rated better than the other methods.

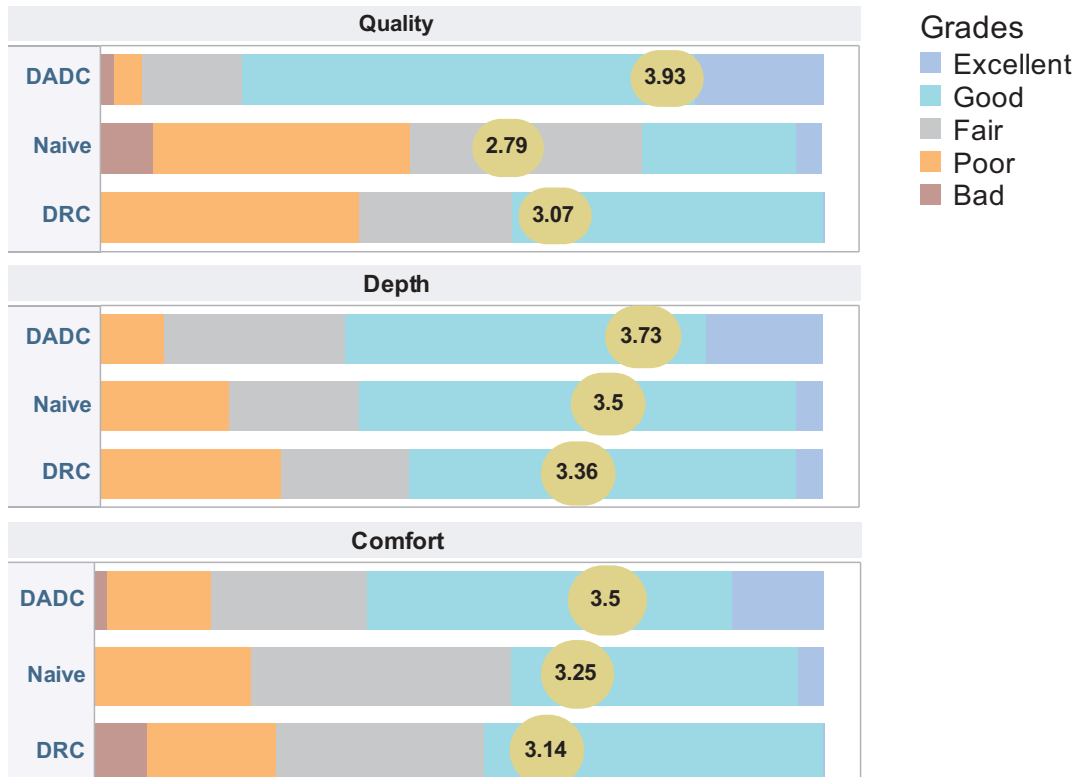


Figure 7.1: Charts describing the subjects' ratings and averages based on 5-point Likert scale for our method and the compared methods. In each chart, the average grade is indicated in a circle.

Figure 7.2 shows results of the preferences collected from the questions comparing our method with other methods described in Section 4. Different from the rating analysis of the methods, this chart shows the preferences in percentages for our method directly in comparison with other two methods. These preferences are determined by the subjects by taking into account image quality, 3D perceived depth, visual comfort and overall quality. The study showed that DADC was preferred in overall quality over the two other methods, both with a 64,28 % preference; whereas in 21,43 % of the cases the Naive method was preferred over ours and 25 % showed preferences of DRC. The high performance of the Naive method is due to the fact that the static disparity levels were initialized compatibly with the scenes, for a fair comparison.

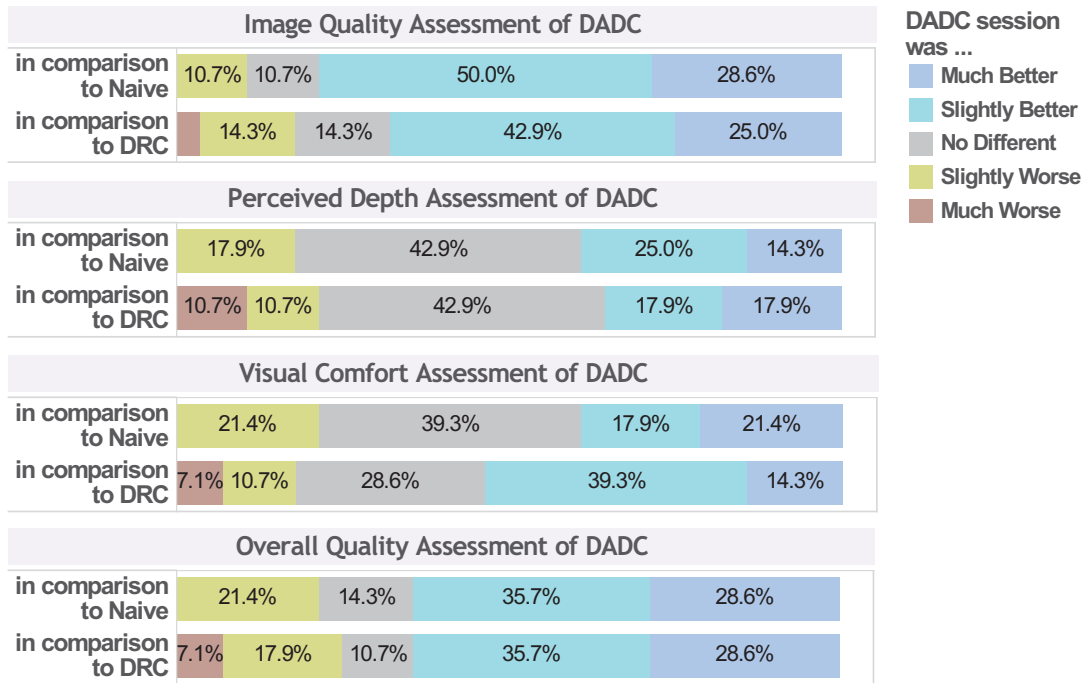


Figure 7.2: Aggregated results from our session comparison questionnaires demonstrating relative user preferences of our DADC method in percentages. Scores are relative to Naive method in the first row and DRC method in the second.

To evaluate the cinematographic quality of each method, we have plotted *the depth charts* [22] of a test sequence illustrating the distribution of the depth budget over time with each method. The charts in Figure 7.3 shows the minimum and maximum depth values of the scene, with respect to the physical display surface (Figure 2.1). The figure also shows the perceived depth of the most salient scene element, which we designated based on the scene and the significance scores (orange curve). The results show that our method achieves the goal of keeping the most significant object closed to the planar screen as much as possible. Based on these results, we can claim that our method prevents the accommodation/convergence conflict to a large extent.

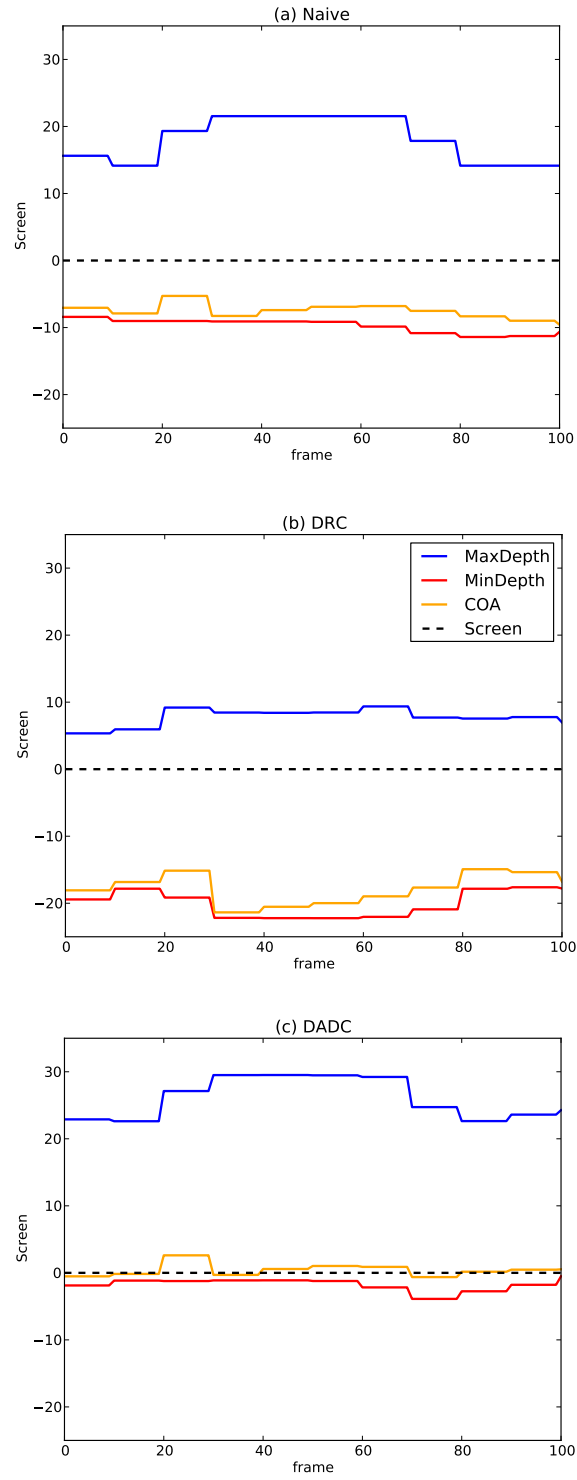


Figure 7.3: Depth charts of an evaluated scene for the first hundred frames with
(a) Naive method (b) DRC (c) DADC

Chapter 8

Conclusion

This part has presented an approach for conveying scene depth in any arbitrary interactive 3D scene content by automatically calculating the stereoscopic camera parameters of convergence and camera separation. Our method specifies a depth configured according to the distribution and importance degree of salient elements in the scene, and automatically finds the parameters for mapping total scene depth to this specified depth range.

This method for stereoscopic camera parameter arrangement allows 3D scene content creators to adjust and distribute available perceived depth in a way that the perceived depth is controlled and limited to the stereoscopic comfort zone of the users and accommodation/convergence conflict is not violated by keeping the focus or the convergence of the camera closer to the elements of interest.

Part III

RBF Interpolation-Based Disparity Refinement

Chapter 9

Radial Basis Function Interpolation

Numerous computer graphics and computer vision problems suggest the involvement of interpolation. Sometimes, the sample locations exist on a regular grid structure. For example, this is normally the situation in the case of an image data as the samples are lined up with respect to a CCD array. A B-spline surface data are also composed on a regular grid structure in parameter space. Spline interpolation methods deal with these cases.

For many other problems of computer graphics, such as surface-reconstruction, surface-deformation, motion interpolation and inverse kinematics, meshless methods for fluid dynamics, and appearance-representation [23]; data locations are not that structured but rather scattered. However, scattered data interpolation methods are not as well-known, e.g. most computer graphics textbooks do not cover these methods. Yet, over the last two decades, scattered data interpolation methods have been increasingly used in computer graphics research.

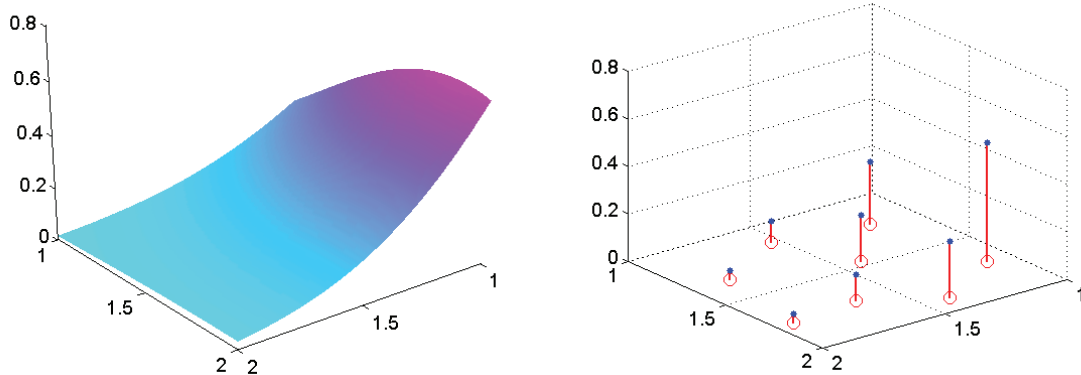


Figure 9.1: Sample data from test function.

9.1 Scattered Data Interpolation

Interpolation is an essential issue that has been contemplated in many fields. One can consider interpolation as an inverse problem because the solution possibly includes many more degrees of freedom than the given limited data sample, as, for example, every point on a curve has with respect to the known points. The interpolation type can be considered as a prior, making the inverse problem solvable.

For the case of scattered data interpolation, it is necessary to perfectly fit the computed function to the data. On the other hand, in scattered data approximation, it is enough when the computed function only merely passes close to the sample data. Therefore, the approximation permits treatment of noisy data. In spite of the fact that scattered data interpolation and scattered data approximation are different problems, a portion of the same algorithms can be applied in both cases.

Scattered data modeling requires an unknown real-valued function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ from a given scattered data vector $f|_X = (f(x_1), f(x_2), \dots, f(x_N))^T \in \mathbb{R}^N$ of function of N values sampled at a finite set $X = \{x_1, x_2, \dots, x_N\} \subset \mathbb{R}^d$ of N

distinct data points. Here, the data points have no underlying structure or order between their relative locations.

For instance, in Figure 9.1 [24], the graph of an unknown function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ is shown on the left side and 8 data points sampled from that function f at scattered data locations are shown on the right side.

An example for such scattered data is the *Hurrungane* data set [25]. This is a set of sampled height values $\{f(x)\}_{x \in X}$, $X \subset \mathbb{R}^2$, of a mountainous area in Norway at $|X| = 23092$ distinct geographic locations. 2D and 3D views of the set are displayed in Figure 9.2. In this case, the goal is to approximate the shape of the mountains' surfaces. One way of doing so is to consider solving the interpolation problem $s|_X = f|_X$, i.e.

$$s(x_k) = f(x_k), \quad \text{for } 1 \leq k \leq N, \quad (9.1)$$

where $s : \mathbb{R}^d \rightarrow \mathbb{R}$ is some suitable function, referred to as *interpolant* to f .

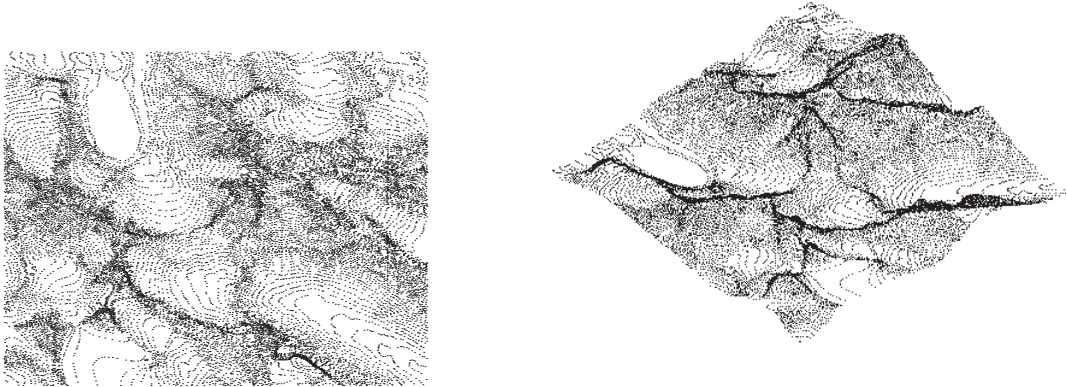


Figure 9.2: 2D view (left) and 3D view (right) of the data set Hurrungane (23092 points).

9.2 Introduction to Radial Basis Functions

Radial basis functions (RBFs) happen to be the most versatile and also one of the most commonly used techniques for scattered data interpolation [26] [27]. From a high level view, an RBF interpolation operates by aggregating a set of replicas of a single basis function. Each of the replicas is centered at a data point and scaled with respect to the interpolation conditions.

In concept, radial basis functions can be likened to fitting a rubber membrane through the given sample data values while minimizing the total curvature of the membrane surface. The selected basis function governs how the membrane will fit through those sample values. Figure 9.3 [28] demonstrates conceptually how a radial basis function surface fits between a set of sample values of elevation. In the cross section (Figure 9.3(c)), it can be noticed that the surface in fact passes through the sample values.

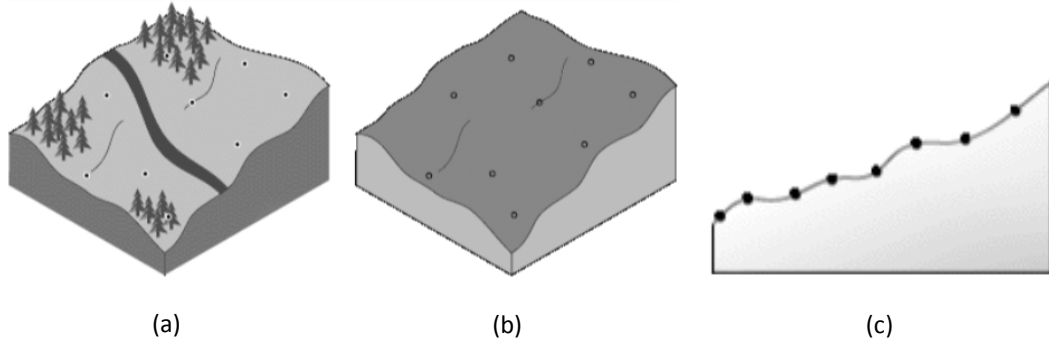


Figure 9.3: Conceptual representation of RBF interpolation (a) Real terrain surface (b) Surface with sampled location indicated (c) Cross-section of the RBF interpolated surface.

The radial basis function interpolation methods, as they are exact interpolators, contrast with both the global polynomial interpolators and the local polynomial interpolators, which are inexact interpolators that do not require the surface to pass through the sample points. In comparison of RBF interpolation to inverse distance weighted interpolation, which is also an exact interpolator, inverse distance weighted interpolation can not predict values above the maximum or below

the minimum sample value, as can be seen in the cross section of sample data in Figure 9.4 [28]; while RBFs can predict values above the maximum and below the minimum sample values.

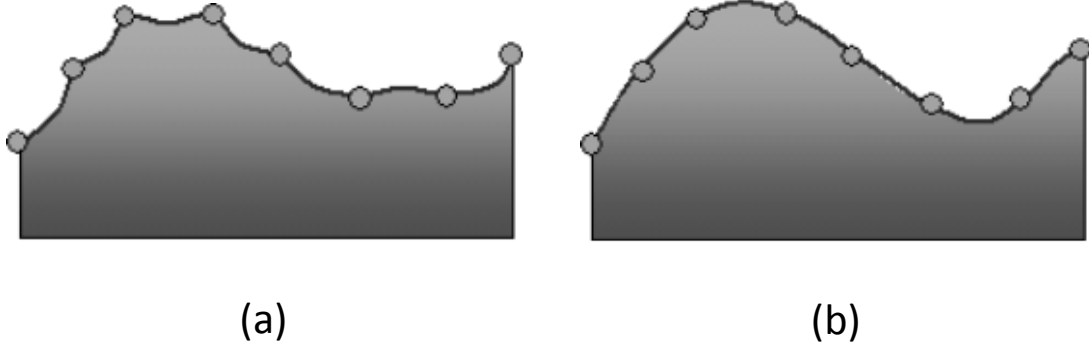


Figure 9.4: Comparison of (a) inverse distance weighted interpolation (b) radial basis function interpolation.

A radial function can be defined as a function that is radially symmetric around a point $\mathbf{x}_c$ which is called the center of that function. For a kernel $K : \mathbb{R}^s \times \mathbb{R}^s \rightarrow \mathbb{R}$ with input vectors $\mathbf{x} = [x_1, x_2, \dots, x_s]^T$ and $\mathbf{x}_c = [(x_c)_1, (x_c)_2, \dots, (x_c)_s]^T$, K is a radial function if it can be designated as $K(\mathbf{x}, \mathbf{x}_c) = \Phi(r)$, where $r = \|\mathbf{x} - \mathbf{x}_c\|$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}_c$. For utilizing $\Phi(r)$ as a radial basis function, the center $\mathbf{x}_c$ is fixed to a constant point and $\mathbf{x}$ is regarded as the input variable. It is noteworthy that the univariate function Φ is independent of s , the number of input dimensions. Consequently, methods involving radial basis functions can, in principle, be conveniently adapted to solve problems of higher dimensions.

Given a scattered data $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ of n distinct data points in $\mathbb{R}^s$ and a corresponding set of n values $y_1, y_2, \dots, y_n$ sampled from an unknown function f such that $y_i = f(x_i)$, we can select a radial function Φ and a set of centers $\{\mathbf{x}_{c_1}, \mathbf{x}_{c_2}, \dots, \mathbf{x}_{c_m}\}$ for some $m \in \mathbb{N}$, in order to form a basis $\{\Phi(\|\cdot - \mathbf{x}_{c_1}\|), \Phi(\|\cdot - \mathbf{x}_{c_2}\|), \dots, \Phi(\|\cdot - \mathbf{x}_{c_m}\|)\}$. So that, this basis can be utilized to build an approximation $\tilde{f}$ of f .

An option is to center a radial basis function on each data site. For that case,

$\forall i \in \{1, 2, \dots, n\}$ there will be one basis function with $\mathbf{x}_c = \mathbf{x}_i$ and $\tilde{f}$ will be built from n RBFs s.t.

$$\tilde{f}(x) = \sum_{j=1}^n w_j \Phi(\|\mathbf{x} - \mathbf{x}_j\|) \quad (9.2)$$

with constant coefficients w_j . In an interpolation, the constants w_j , which are called the *weights*, are defined by guaranteeing that $\tilde{f}$ matches the given scattered data $\mathbb{X}$, $\forall \mathbf{x}_i$. This is achieved by imposing $\tilde{f}(x_i) = y_i$, which results in the following system of linear equations

$$K\mathbf{w} = \mathbf{y}, \quad (9.3)$$

where

$$K = \begin{bmatrix} \Phi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \Phi(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \dots & \Phi(\|\mathbf{x}_1 - \mathbf{x}_n\|) \\ \Phi(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \Phi(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \dots & \Phi(\|\mathbf{x}_2 - \mathbf{x}_n\|) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi(\|\mathbf{x}_n - \mathbf{x}_1\|) & \Phi(\|\mathbf{x}_n - \mathbf{x}_2\|) & \dots & \Phi(\|\mathbf{x}_n - \mathbf{x}_n\|) \end{bmatrix}, \quad (9.4)$$

$\mathbf{w} = [w_1, w_2, \dots, w_n]^T$, and $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$

The coefficients w_j can be computed by solving the above linear system. One also needs to define Φ , i.e. a RBF.

9.2.1 Basis Functions

Applicable methods for solving Eq. 9.3 and whether such a solution even exists are determined by the choice of basis functions $\Phi(r)$, where $r = \|\mathbf{x} - \mathbf{x}_c\|$.

Whether the interpolated function is smooth, whether the function attenuates to

zero away from the sample data, and whether the function overshoots are among the top issues due for consideration for many of the applications.

Naturally, it is not achievable to obtain both smoothness and lack of overshoot, or both smoothness and attenuation to zero away from the sample data at the same time. Yet, these criteria can be combined with desired strengths.

Although there is not yet a general characterization for functions that are suitable as RBF kernels [29], positive-definite functions are among those that will generate a non-singular K matrix for any choice of data locations.

The linear system $K\mathbf{w} = \mathbf{y}$ has a unique solution if the interpolation matrix K is a symmetric positive definite matrix. A positive definite matrix is invertible since all of its eigenvalues are positive. Furthermore, there exists methods such as Cholesky factorization which can be utilized in order to solve a linear system that is symmetric positive definite more efficiently than methods devised for a general linear system [30].

9.2.1.1 Families of Basis Functions

It is vital to select an appropriate radial basis kernel $\Phi(r)$ family among the available many. Below are the most prominent families of radial basis functions in the literature [31]:

- Gaussian:

$$\Phi(r) = \exp(-r^2/r_0^2) \quad (9.5)$$

The main benefit of a Gaussian basis function is that it is compact, i.e., it is small at distances about $3r_0$ from the center, where r_0 is the *scale factor* as we adopt to call it; although it has also been called as *shape parameter*, *width parameter* and *model radius* in the literature. The Gaussian basis function can be considered zero for distances of $6r_0$ or larger from the center. Consequently, it results in a linear system with a sparse matrix. The scale factor

r_0 will make model sharp and non-smooth if it is too small (Figure 9.5), and a too large r_0 leads to a linear system with an ill-conditioned matrix.

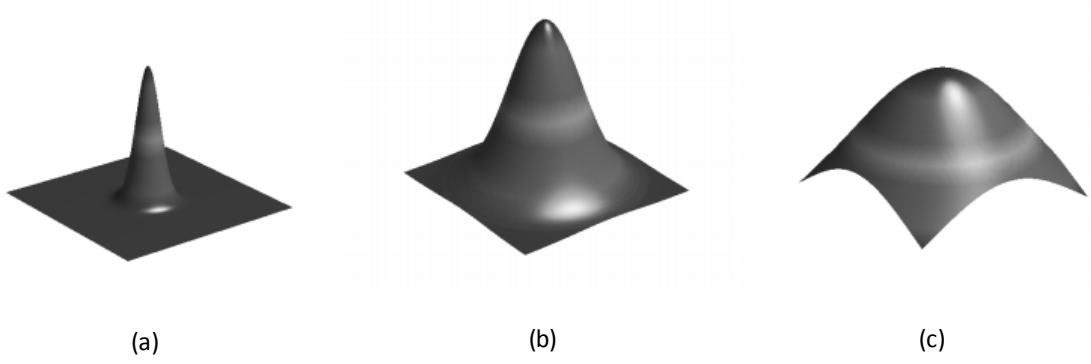


Figure 9.5: Gaussian RBFs plotted with (a) $r_0 = 1/3$ (b) $r_0 = 1$ (c) $r_0 = 2.5$, on the same domain.

- Multiquadric:

$$\Phi(r) = \sqrt{1 + (r/r_0)^2} \quad (9.6)$$

Multiquadric basis functions are non-compact, i.e., non-zero at any point. This poses a major drawback as one needs to solve systems of dense matrices.

- Inverse multiquadric:

$$\Phi(r) = 1/\sqrt{1 + (r/r_0)^2} \quad (9.7)$$

It is found [32] that excellent approximations can be provided by inverse multiquadratic radial basis functions. This is even so when the number of centers is small.

The multiquadric and inverse multiquadric basis functions are both part of the generalized multiquadric family of RBFs which are defined by $\Phi(r) = (1 + (r/r_0)^2)^\beta$. An in-depth analysis on generalized multiquadrics and the parameter β can be found in [33].

- Polyharmonic spline:

$$\Phi(r) = \begin{cases} r^k \ln(r) & \text{with } k = 2, 4, 6, \dots \\ r^k & \text{otherwise} \end{cases} \quad (9.8)$$

Polyharmonic spline basis functions are also non-compact basis functions. This drawback very often makes this basis function family inappropriate for medium to large scale problems.

- Thin-plate spline:

$$\Phi(r) = r^2 \log(r) \quad (9.9)$$

Thin-plate spline is a special case of polyharmonic spline. The usefulness of thin-plate spline for fitting smooth functions of two variables and, more generally, even number of variables is demonstrated in [32] and [34].

Chapter 10

Approach

The overview of the RBF interpolation-based disparity refinement approach is presented in Figure 10.1. The framework for the approach necessitates a specific type of content-preparation phase as a preliminary. Once the content is ready to be processed by the framework, the main phase takes over. There are two alternating parts to the main phase. These are named as the Play Mode and the Edit Mode of the framework. Here, assuming that the content is prepared so that it has a balanced distribution of camera-parameters-assigned scene elements and those parameters are well-adjusted, the Edit Mode stands to be simply optional.

10.1 Content-Preparation

The approach requires that the scene contents of the given virtual medium include at least a certain number of scene elements that have their stereoscopic camera parameters assigned by the user, who, in this case, acts as the content-creator and/or the content-editor. We set the threshold for the number of such scene elements to be 5 at minimum. It is also assumed that these parameter-assigned scene elements are among the most salient within the given environment and that they are distributed sparsely within that environment. In the content-preparation phase, the user sets the required stereoscopic camera parameters for any scene

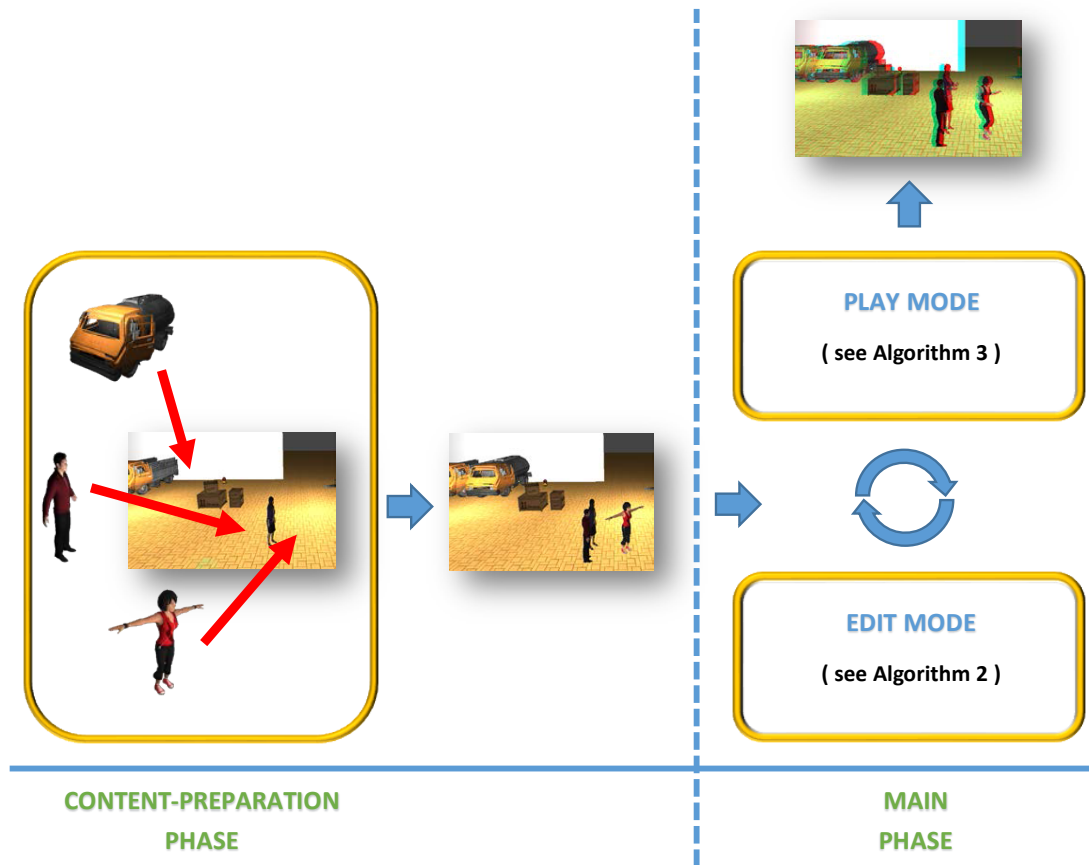


Figure 10.1: Overview of the approach is presented. The framework for the approach necessitates a specific type of content-preparation phase as a preliminary. Once the content is ready to be processed by the framework, the main phase takes over. There are two alternating parts to the main phase. These are named as the Play Mode and the Edit Mode of the framework.

element they choose to. These parameters are the interaxial separation of the two stereo camera pair (t_c), the convergence distance where the two cameras meet (Z_c), and the focal length of the virtual lens of the cameras (f). The creator/editor is supposed to shape the, so called, overall *depth narrative* of the scene by adjusting the set of these parameters per scene element in accordance with the consideration of the comfortable viewing issues for the given environment.

10.2 Main Phase

The main phase of the approach can be run in any given virtual environment as long as the environment meets the prerequisites listed above. The environment can be either interactive or not. Regardless, the method runs online in both cases. The main phase works in two alternating modes, namely, the Play Mode and the Edit Mode. Play Mode, as the name implies, is the default state of the system where the user experiences the uninterrupted stereo 3D output of our approach as they go through the virtual environment. Edit Mode is optional yet vital. It is complementary to the Play Mode as it can be selected to enhance the total disparity further or to remedy the shortcomings of a poorly established content. Let us introduce the Edit Mode first.

10.2.1 Edit Mode

Overall mechanism of Edit Mode is outlined in Algorithm 2. Edit Mode essentially serves an auxiliary purpose to fine-tune the stereo camera parameters, as to infuse the overall depth feeling with a personal touch. Yet, with it, the user is also able to create (or re-create) the depth narrative of the whole scene from scratch, as well. We believe, for the most of the scenarios, the user can select the Edit Mode right after entering into the new scene, e.g., as soon as the new level is loaded in a game, use the Edit Mode as an initial tune-up and then be done with editing at least till the next scene.

Algorithm 2 Edit Mode Algorithm

```
1: procedure EDITMODE(  $O[]$ ,  $SP(x, y, z)$ ,  $(C_x, C_y, C_z)$ ,  $(M_i, M_j)$  )
2:            $\triangleright O[]$  is the list of parameter-assigned scene-elements.
3:            $\triangleright SP(x, y, z)$  is the parameter-stack for the whole virtual scene.
4:            $\triangleright (C_x, C_y, C_z)$  is position of the main virtual camera.
5:            $\triangleright (M_i, M_j)$  is position of the mouse cursor on the application window.
6:
7:   while  $\neg PlayModeRequest$  do
8:
9:     if  $Raycast((C_x, C_y, C_z), (M_i, M_j)) \neq NULL$  then
10:       $\triangleright$  Raycasting from  $(C_x, C_y, C_z)$  in the direction of  $(M_i, M_j)$ 
11:      if  $Raycast((C_x, C_y, C_z), (M_i, M_j)) \in O[]$  then
12:         $\triangleright$  If the raycast hits a scene-element,
13:         $\triangleright$  that element is assigned to  $hit$ .
14:         $hit \leftarrow Raycast((C_x, C_y, C_z), (M_i, M_j))$ 
15:      else
16:         $hit \leftarrow NULL$ 
17:      end if
18:    end if
19:
20:    if  $ShowParameterAdjustmentMenu$  and  $hit \neq NULL$  then
21:       $ShowMenu(SP(C_x, C_y, C_z))$ 
22:       $\triangleright$  Parameter adjustment menu is shown upon user request.
23:       $\triangleright$  User can now change the parameters at  $(C_x, C_y, C_z)$ .
24:      if  $ParameterAssignmentRequested$  then
25:         $hit.SP \leftarrow SPmenu.SP$ 
26:         $\triangleright$  Upon user request, the adjusted parameter set on the menu
27:         $\triangleright$  is assigned to the scene element pointed by  $hit$ .
28:      end if
29:    end if
30:
31:  end while
32: end procedure
```

We aimed to devise an intuitively-designed user-interface for the Edit Mode. So that, even the most novice user can quickly learn it and manage to utilize it without hassle. Some sample screenshots of the Edit Mode user-interface can be seen in Figure 10.2.

During the Edit Mode, the user is able to roam the virtual scene freely in a first-person-camera style by translating and rotating the main camera (Figure 10.2a). The main camera is cloned to form the stereoscopic virtual camera pair. On the screen, the user sees the stereo scene view created by this pair. The left and the right cameras are distanced by half of the interaxial-separation (t_c) from the position of the main camera to the left and the right, respectively. The rotation of the main camera is also reflected on the stereo camera pair but with the adjustment so that the camera image capturing axes meet at the convergence-distance (Z_c) in front of the main camera position. The focal length (f) of the main camera and the virtual stereo camera pair are the same.

The user can pick any scene element within the stereo scene view. Picking is achieved by raycasting from the camera in the direction pointed by the mouse cursor on the application window. The hit indicator label on the top left corner of the window (Figure 10.2b) shows the name of the nearest element hit by the raycast. The user can pick the element by locking the raycast selection by pressing the assigned keyboard shortcut (the default is the tab key). Once the element is picked, the stereo camera parameter adjustment menu is displayed (Figure 10.2c). Then, the user can adjust t_c , Z_c and f using the slider for each one on the menu. Since the sliders directly control the stereo camera pair, the adjusted values are instantly reflected on the view. Upon finishing the adjustment, the user can replace the parameter set of the picked scene element with the newly adjusted values by a keyboard shortcut (the default is the space key). When the user is done with adjustments, they can return to the Play Mode instantly.



(a) User roaming the camera about the scene.



(b) User designates a scene-element.



(c) Stereo camera parameter adjustment menu shown upon locking onto a scene-element.

Figure 10.2: Sample screen-captures of the user interface from an Edit Mode user session.

10.2.2 Play Mode

Play Mode is the default mode of the main phase as it is the main stage of our approach. As its process is outlined in Algorithm 3, the method produces the virtual stereoscopic camera parameters t_c , Z_c and f online while continuously scanning the virtual scene for changes in the variables, namely, the number of parameter-assigned scene elements, the distribution of those scene-elements in the scene with respect to the camera, and, most importantly, their assigned parameters, that affect the algorithm. The virtual stereo camera pair is continuously updated with the computed stereo camera parameters.

As long as the user is content with the overall perceived disparity of the scene, the method runs in the Play Mode from start to finish, i.e., till the user is done with that scene. Otherwise, the user can interrupt the Play Mode and enter the Edit Mode in order to shape the depth feeling according to their liking any time they may wish so.

During the Play Mode run, for each of the three parameters, a separate set of RBF interpolation weights $w(i)$ are found by solving the the set of n equations

$$F_d(j) = \sum_{i=1}^n w(i) \Phi(\|x_d(i) - x_d(j)\|), \quad (10.1)$$

where n is the number of stereoscopic-camera parameter assigned scene-elements, $x_d(i)$ is the position of the data point i (i.e., the scene element i), $\Phi(\cdot)$ is the designated RBF family, and $F_d(i)$ is the value of that parameter at the data point i . The user is able to change the RBF family $\Phi(\cdot)$ in use to any one of the available alternatives at any moment during the Play Mode run.

Algorithm 3 Play Mode Algorithm

```
1: procedure PLAYMODE(  $O[]$ ,  $SP(x, y, z)$ ,  $(C_x, C_y, C_z)$  )
2:
3:            $\triangleright O[]$  is the list of parameter-assigned scene-elements.
4:            $\triangleright SP(x, y, z)$  is the parameter-stack for the whole virtual scene.
5:            $\triangleright (C_x, C_y, C_z)$  is position of the main virtual camera.
6:
7:   while  $\neg \text{EditModeRequest}$  do
8:
9:            $\triangleright$  Separately for  $t_c$ ,  $Z_c$  and  $f$ :
10:           $w_{t_c}[] = \text{RBFweights}(O[].xyz, O[].SP.t_c, \Phi, r_0)$ 
11:           $w_{Z_c}[] = \text{RBFweights}(O[].xyz, O[].SP.Z_c, \Phi, r_0)$ 
12:           $w_f[] = \text{RBFweights}(O[].xyz, O[].SP.f, \Phi, r_0)$ 
13:           $\triangleright$  RBF weights are solved according to current positions and
14:           $\triangleright$  stereo-parameters of the scene elements
15:           $\triangleright$  for a family of radial basis function  $\Phi$ 
16:           $\triangleright$  and a scale factor  $r_0$  specified by the user.
17:
18:           $\triangleright$  Again for each of the three parameters:
19:           $SP(C_x, C_y, C_z).t_c = \text{RBFinterp}(w_{t_c}[], (C_x, C_y, C_z), O[].xyz, \Phi, r_0)$ 
20:           $SP(C_x, C_y, C_z).Z_c = \text{RBFinterp}(w_{Z_c}[], (C_x, C_y, C_z), O[].xyz, \Phi, r_0)$ 
21:           $SP(C_x, C_y, C_z).f = \text{RBFinterp}(w_f[], (C_x, C_y, C_z), O[].xyz, \Phi, r_0)$ 
22:           $\triangleright$  RBF-interpolated values are found
23:           $\triangleright$  at the current camera position  $(C_x, C_y, C_z)$ 
24:           $\triangleright$  for the same family of radial basis function  $\Phi$ 
25:           $\triangleright$  and the scale factor  $r_0$ .
26:
27:   end while
28: end procedure
```

Then, for a location x at where there is originally no parameter data, the RBF interpolated value $F(x)$ of that parameter is computed by

$$F(x) = \sum_{i=1}^n w(i) \Phi(\|x - x_d(i)\|) \quad (10.2)$$

with respect to the computed weights $w(i)$ and the chosen RBF family $\Phi(\cdot)$.

10.3 System

The entire system for the approach, including the user-interface, is put together using the Unity development environment. For the radial basis function interpolation-based computation of the stereoscopic camera parameters t_c , Z_c and f , we utilized the C++ library written by John Burkardt [35] as a plugin to the system. This library is chosen because of its lightweight and robust implementation of fast radial basis function interpolation which provided online operation of the framework. The library, by default, supports the use of Gaussian, multiquadratic, inverse-multiquadratic and thin-plate spline radial basis function families in addition to the availability of user-defined scale factor r_0 .

10.4 Testing

We tested the method in two different settings. The first setting is a non-interactive virtual environment. Here, the camera travels along one of the pre-defined paths in the virtual environment. The paths are surrounded by scene-elements of varying significance which are assigned stereoscopic-camera parameter sets that output varying types of stereo narratives. A sample path can be seen in Figure 10.3.

The second setting is an interactive environment. In this setting, the user can

control the camera in 3D space freely.

Although the user has the option to select one of the several radial basis kernel families, we mainly employed Gaussian kernels (Equation 9.5) during testing since the compactness of the Gaussian basis functions gives us high performance.

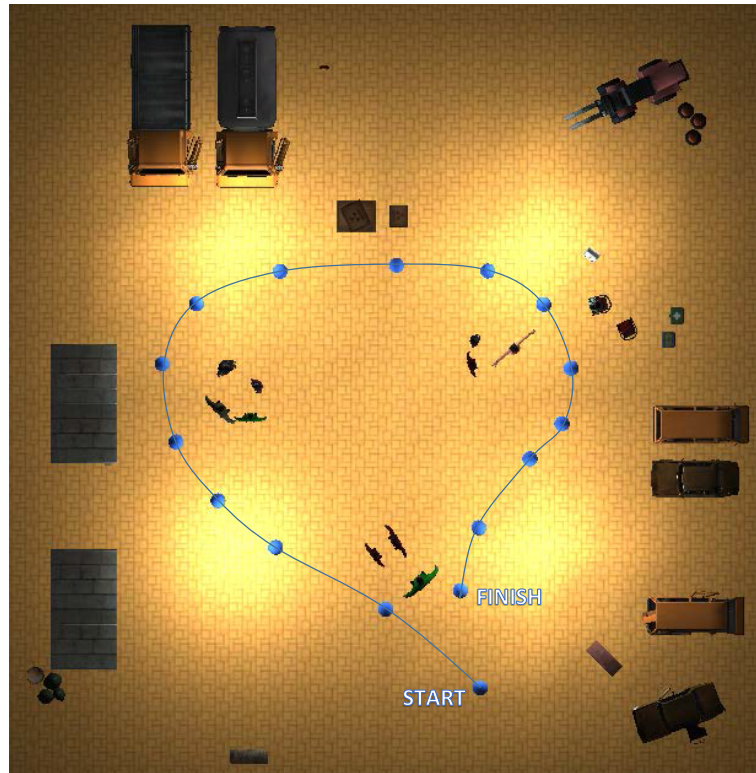


Figure 10.3: Diagram of a sample path used in non-interactive testing of the approach.

The user also has the option to choose between the available two different stereo-output settings, which are red-cyan anaglyph output and side-by-side output, during the Play Mode via the user-interface. A sample for each output type can be seen in Figure 10.4.



Figure 10.4: Available stereo-output settings of the framework (a) red-cyan anaglyph output (b) side-by-side output (before being overlapped by the display device).

The testing demonstrated that the proposed method works reasonably well in both interactive and non-interactive settings with the help of a meticulously-adjusted scene, by which we mean a scene for which the stereoscopic camera parameters are well-adjusted for every scene-element. It is observed that the method is unable to produce decent output at regions where the parameter assigned scene elements are scarce.

Chapter 11

Conclusion

An approach for radial basis function interpolation-based automatic adjustment of stereo camera parameters for virtual environments has been presented in this part of our work. The method is observed to work well providing scene-wide depth narrative that is in line with the creative choices of the content producer under both interactive and non-interactive settings. The method also allows for remedying the possible shortcomings of the initial disparity setup by having an editing mode through which each user can personalize their own experience of the system by fine-tuning the parameters to their liking and visual comfort.

In addition to the informal testing we executed, we plan to conduct a formal user study, as we did in the previous part, to further validate the quality of the method and to detect shortfalls of it, if any.

In order to study the effects of the radial basis kernel used in interpolation on the method outcome further, we need to conduct experiments with other kernel families than the Gaussian kernel we used during testing. This also calls for experimenting with the scale factor r_0 as needed.

As future work, we will look into how we can extend the method to better handle the so-called accommodation-convergence conflict. We will also explore ways of

combining the method presented here with the Dynamic Attention-Aware Disparity Control method described in the previous part with the goal of achieving even superior visual comfort and depth feeling in a standalone system.

Bibliography

- [1] A. Shamir and O. Sorkine, “Visual media retargeting,” in *ACM SIGGRAPH ASIA 2009 Courses*, SIGGRAPH ASIA '09, pp. 11:1–11:13, ACM, 2009.
- [2] P. Didyk, T. Ritschel, E. Eisemann, K. Myszkowski, H.-P. Seidel, and W. Matusik, “A luminance-contrast-aware disparity model and applications,” *ACM Trans. Graph.*, vol. 31, pp. 184:1–184:10, Nov. 2012.
- [3] M. Lang, A. Hornung, O. Wang, S. Poulakos, A. Smolic, and M. Gross, “Nonlinear disparity mapping for stereoscopic 3D,” *ACM Trans. Graph.*, vol. 29, pp. 75:1–75:10, July 2010.
- [4] T. Oskam, A. Hornung, H. Bowles, K. Mitchell, and M. Gross, “OSCAM-optimized stereoscopic camera control for interactive 3D,” in *SA'11 Proceedings of the 2011 SIGGRAPH Asia Conference*, vol. 30, p. 189, ACM New York, 2011.
- [5] I. P. Howard and B. J. Rogers, “Seeing in depth, volume 2: Depth perception,” *Toronto: I Porteous*, 2002.
- [6] P. Milgram and M. Krüger, “Adaptation effects in stereo due to on-line changes in camera configuration,” *Proc. SPIE Vol. 1669-13, Stereoscopic Displays and Applications III*, 1992.
- [7] B. Mendiburu, *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*. Focal Press, 2009.
- [8] F. Zilly, J. Kluger, and P. Kauff, “Production rules for stereo acquisition,” *Proceedings of the IEEE*, vol. 99, no. 4, pp. 590–606, 2011.

- [9] F. Zilly, M. Muller, P. Kauff, and R. Schafer, "STAN - An assistance system for 3D productions: From bad stereo to good stereo," in *Electronic Media Technology (CEMT), 2011 14th ITG Conference on*, pp. 1–6, IEEE, 2011.
- [10] S. Heinzle, P. Greisen, D. Gallup, C. Chen, D. Saner, A. Smolic, A. Burg, W. Matusik, and M. Gross, "Computational stereo camera system with programmable control loop," *ACM Trans. Graph.*, vol. 30, pp. 94:1–94:10, August 2011.
- [11] S. Koppal, C. Zitnick, M. Cohen, S. B. Kang, B. Ressler, and A. Colburn, "A Viewer-Centric Editor for 3D Movies," *Computer Graphics and Applications, IEEE*, vol. 31, pp. 20–35, jan.-feb. 2011.
- [12] P. Didyk, T. Ritschel, E. Eisemann, K. Myszkowski, and H.-P. Seidel, "A perceptual model for disparity," *ACM Transactions on Graphics (Proceedings SIGGRAPH 2011, Vancouver)*, vol. 30, no. 4, 2011.
- [13] G. R. Jones, D. Lee, N. S. Holliman, and D. Ezra, "Controlling perceived depth in stereoscopic images.," SPIE, 2001.
- [14] U. Celikcan, G. Cimen, E. B. Kevinc, and T. Capin, "Attention-aware disparity control in interactive environments," *The Visual Computer*, vol. 29, no. 6-8, pp. 685–694, 2013.
- [15] M. Guttman, L. Wolf, and D. Cohen-Or, "Semi-automatic stereo extraction from video footage," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 136–142, IEEE, 2009.
- [16] A. Greß, M. Guthe, and R. Klein, "GPU-based Collision Detection for Deformable Parameterized Surfaces," in *Computer Graphics Forum*, vol. 25, pp. 497–506, Wiley Online Library, 2006.
- [17] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Reviews Neuroscience*, vol. 2, no. 3, pp. 194–203, 2001.
- [18] M. Reddy, "Perceptually optimized 3D graphics," *Computer Graphics and Applications, IEEE*, vol. 21, pp. 68–75, sep/oct 2001.

- [19] T. P. Runarsson and X. Yao, “Stochastic ranking for constrained evolutionary optimization,” *Evolutionary Computation, IEEE Transactions on*, vol. 4, no. 3, pp. 284–294, 2000.
- [20] S. Johnson, “The NLOpt nonlinear-optimization package,(2011),” *URL* <http://ab-initio.mit.edu/nlopt>.
- [21] “ITU-R Recommendation BT.2021, Subjective methods for the assessment of stereoscopic 3DTV systems,” 2012.
- [22] C.-W. Liu, T.-H. Huang, M.-H. Chang, K.-Y. Lee, C.-K. Liang, and Y.-Y. Chuang, “3D cinematography principles and their applications to stereoscopic media processing,” in *Proceedings of the 19th ACM international conference on Multimedia, MM ’11*, (New York, NY, USA), pp. 253–262, ACM, 2011.
- [23] J. Lewis, F. Pighin, and K. Anjyo, “Scattered data interpolation for computer graphics,” *SIGGRAPH Asia Course*, 2010.
- [24] M. Mongillo, “Choosing basis functions and shape parameters for radial basis function methods,” *SIAM Research*, 2011.
- [25] A. Iske, E. Quak, and M. S. Floater, *Tutorials on multiresolution in geometric modelling: summer school lecture notes*. Springer, 2002.
- [26] M. J. Powell, “The theory of radial basis function approximation in 1990,” *Advances in numerical analysis*, vol. 2, pp. 105–210, 1992.
- [27] M. J. Powell, “Radial basis functions for multivariable interpolation: a review,” in *Algorithms for approximation*, pp. 143–167, Clarendon Press, 1987.
- [28] <http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm>. [Online; accessed 14-January-2015].
- [29] G. F. Fasshauer, *Meshfree approximation methods with MATLAB*. World Scientific Publishing Co., Inc., 2007.
- [30] L. N. Trefethen and D. Bau III, *Numerical linear algebra*, vol. 50. Siam, 1997.

- [31] B. J. C. Baxter, *The interpolation theory of radial basis functions*. PhD thesis, University of Cambridge, 1992.
- [32] R. Franke, “Scattered data interpolation: Tests of some methods,” *Mathematics of computation*, vol. 38, no. 157, pp. 181–200, 1982.
- [33] M. E. Chenoweth and S. A. Sarra, “A numerical study of generalized multi-quadric radial basis function interpolation,” *SIAM Research*, vol. 2, pp. 58–70, 2009.
- [34] M. D. Buhmann, “Multivariate cardinal interpolation with radial-basis functions,” *Constructive Approximation*, vol. 6, no. 3, pp. 225–255, 1990.
- [35] “Radial Basis Function Interpolation in 2D.” http://people.sc.fsu.edu/~jburkardt/cpp_src/rbf_interp_2d/rbf_interp_2d.html. [Online; accessed 14-January-2015].