Free View Rendering for 3D Video

Edge-Aided Rendering and Depth-Based Image Inpainting

Suryanarayana M. Muddala

Mittuniversitetet
MID SWEDEN UNIVERSITY

Department of Information and Communication Systems
Faculty of Science, Technology and Media
Mid Sweden University

Doctoral Thesis No. 226
Sundsvall, Sweden
2015

Mittuniversitetet
Informations och kommunikationssystem
SE-851 70 Sundsvall
SWEDEN

ISBN 978-91-88025-30-2
ISSN 1652-893X

Akademisk avhandling som med tillstånd av Mittuniversitetet framlägges till offentlig granskning för avläggande av teknologie doktorsexamen torsdagen den 18 juni 2015 i L111, Mittuniversitetet, Holmgatan 10, Sundsvall.

Tryck: Tryckeriet Mittuniversitetet

To My Parents
To My Wife
To My Brother and My Sister
To My Friends

Abstract

Three Dimensional Video (3DV) has become increasingly popular with the success of 3D cinema. Moreover, emerging display technology offers an immersive experience to the viewer without the necessity of any visual aids such as 3D glasses. 3DV applications, Three Dimensional Television (3DTV) and Free Viewpoint Television (FTV) are auspicious technologies for living room environments by providing immersive experience and look around facilities. In order to provide such an experience, these technologies require a number of camera views captured from different viewpoints. However, the capture and transmission of the required number of views is not a feasible solution, and thus view rendering is employed as an efficient solution to produce the necessary number of views. Depth-image-based rendering (DIBR) is a commonly used rendering method. Although DIBR is a simple approach that can produce the desired number of views, inherent artifacts are major issues in the view rendering. Despite much effort to tackle the rendering artifacts over the years, rendered views still contain visible artifacts.

This dissertation addresses three problems in order to improve 3DV quality: 1) How to improve the rendered view quality using a direct approach without dealing each artifact specifically. 2) How to handle disocclusions (a.k.a. holes) in the rendered views in a visually plausible manner using inpainting. 3) How to reduce spatial inconsistencies in the rendered view. The first problem is tackled by an edge-aided rendering method that uses a direct approach with one-dimensional interpolation, which is applicable when the virtual camera distance is small. The second problem is addressed by using a depth-based inpainting method in the virtual view, which reconstructs the missing texture with background data at the disocclusions. The third problem is undertaken by a rendering method that firstly inpaint occlusions as a layered depth image (LDI) in the original view, and then renders a spatially consistent virtual view.

Objective assessments of proposed methods show improvements over the state-of-the-art rendering methods. Visual inspection shows slight improvements for intermediate views rendered from multiview videos-plus-depth, and the proposed methods outperforms other view rendering methods in the case of rendering from single view video-plus-depth. Results confirm that the proposed methods are capable of reducing rendering artifacts and producing spatially consistent virtual views.

In conclusion, the view rendering methods proposed in this dissertation can support the production of high quality virtual views based on a limited number of input views. When used to create a multi-scopic presentation, the outcome of this dissertation can benefit 3DV technologies to improve the immersive experience.

Keywords: 3DV, 3DTV, FTV, view rendering, depth-image-based rendering, hole-filling, disocclusion filling, inpainting, texture synthesis, view synthesis, layered depth image.

Sammanfattning

Tredimensionell Video (3DV) har blivit alltmer populärt, och rönt framgångar inom bland annat 3D-bio. Ny 3D-skärmteknik ger uppslukande upplevelser för betraktaren utan att några visuella hjälpmedel såsom 3D-glasögon krävs. Tredimensionell Television (3DTV) och Free Viewpoint TV (FTV) är lovande tekniker för framtidens hemmabioanläggningar där betraktaren omsluts av den uppslukande upplevelser och ges möjlighet att titta-runt i scenen och själv bestämma bildvinkel. För att åstadkomma detta krävs ett flertal kameravyer tagna ur olika vinklar. Att överföra den mängd kameravyer som är nödvändigt är inte en genomförbar lösning på grund av bandbreddbegränsningar varför rendering används som en effektiv lösning för att
framställa det erforderliga antalet vyer. Djupbildbaserad rendering (Depth Image Based Rendering - DIBR) är en vanligt förekommande renderingsmetod. även fast DIBR är en enkel metod som kan producera önskat antal vyer, medför renderingen ett antal framträdande felkällor, så kallade artefakter. Trots att stora forskningsansträngningar lagts ner under årens lopp för att ta itu med det som skapar dessa artefakter, så finns återstår fortfarande synliga fel att åtgärda.

Denna avhandling behandlar tre problem i syfte att förbättra 3DV kvalitet: 1) Hur kan man förbättra den återgivna visade kvaliteten med hjälp av en direkt metod utan att behandla varje artefakt specifikt. 2) Hur kan man hantera de hål som i bilderna vid DIBR på ett visuellt rimligt sätt med hjälp av ifyllnad (inpainting). 3) Hur man kan minska de inkonsekvenser som uppträder i den renderade vyns bildinformation. Det första problemet tacklas med hjälp av en kant-stödd renderingsmetod som använder en direkt strategi med endimensionell interpolering, och som är tillämplig när avståndet till den virtuella kameravyn från tillgänglig bildinformation är litet. Det andra problemet åtgärdas genom att använda en djupbaserad ifyllnadsmetod i den virtuella vyn, som rekonstruerar den saknade strukturen med hjälp av den bakgrundsdata som finns runt respektive hål. Det tredje problemet löses genom en konverteringsmetod som går ut på att först fylla i de av förgrund täckta områdena i den ursprungliga vyn med en skiktad djupbild (Layered Depth Image - LDI) i den ursprungliga vyn, och sedan använda informationen i LDI för att rendera en virtuell kameravy med konsekvent bildinformation.

Objektiva mätningar av de metoder som föreslås i den här avhandlingen visar förbättringar jämfört med state-of-the-art metoder för rendering. Visuell inspektion visar små förbättringar för kameravyer renderade från multivyformat (Multiview plus depth - MVD), och de föreslagna metoderna överträffar andra renderingsmetoder vid rendering från vy-djupformat (video + depth -V+D). De presenterade resultaten bekräftar att de föreslagna metoderna kan minska renderingsartefakter och producera rumsligt konsekventa virtuella kameravyer.

Sammanfattningsvis visas att de renderingsmetoder som föreslås i denna avhandling stödjer produktionen av virtuella kameravyer av hög kvalitet utifrån ett begränsat antal ingångsvyer. När detta används för att skapa en multiskopisk presentation, kan resultatet av denna avhandling vara till nytta för att förbättra 3DV tekniker som syftar till att förstärka en uppslukande 3D-upplevelse.

Contents

Acknowledgements

I would to express my deepest gratitude to my main supervisor Prof. Mårten Sjöström for providing me an opportunity to pursue PhD studies. Special thanks to my co supervisor Dr. Roger Olsson for his outstanding support during the last four years. Thanks also for their excellent support and necessary help making this thesis possible. In addition I would like to specially acknowledge their unconstrained guidance and interesting questions in all research matters, trust and confidence on me. And even more for their excellent suggestions and support during complicated situations.

Thanks to Dr. Aron Larsson and Prof. Kjell Brunnström for reviewing this thesis. Thanks to all members of our current and former Realistic 3D research group. A wonderful time working with you. Special thanks to Mitra Damghanian, who always helped me in tough times. Thanks to Yun Li and Sebastian Schwartz for their help and discussions. Moreover, I thank all members of the Department of Information and Communication Systems at Mid Sweden University, especially Annika Berggren for managerial support in many occasions. I would also thanks to funders EU European Regional Development Fund, Mellersta Norrland and Länsstyrelsen Västernorrland.

I would like to thank my friends Manikiran Pannem, Rudravarma Batta, Raghavendra Yarlagadda and Anil Kumar Chinni, who helped me to come to this country. Many thanks to Kavitha Mekala and Vamsi Kiran Adhikarla who are with me in critical situations with their suggestions. Special thanks to my friends who supported me during stressful times.

Finally, I want to give special thanks to my parents Narasimha Murthy and Ramalakshmi and my wife Lalitha for their love and support in all circumstances. This thesis would not have been done without you.

List of Papers

This dissertation is based on the following papers herein referred by their Roman numerals:

I S. M. Muddala and M. Sjöström and R. Olsson. Edge-Preserving Depth-Image-Based Rendering Method. In *International Conference on 3D Imaging, Liége, Belgium*, 2012.

II S. M. Muddala and M. Sjöström and R. Olsson and S. Tourancheau. Edge-aided virtual view rendering for multiview video plus depth. In *3D Image Processing (3DIP) and Applications, IS&T/SPIE, Burlingame, CA, USA*, 2013.

III S. M. Muddala and R. Olsson and M. Sjöström. Disocclusion Handling Using Depth-Based Inpainting. In *The Fifth International Conferences on Advances in Multimedia, Venice, Italy*, 2013.

IV S. M. Muddala and R. Olsson and M. Sjöström. Depth-Included Curvature Inpainting for Disocclusion Filling in View Synthesis. In *International Journal On Advances in Telecommunications, volume 6, issue3&4, pages 132-142*, 2013.

V S. M. Muddala and M. Sjöström and R. Olsson. Depth-based inpainting for disocclusion filling. In *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), Budapest, Hungary*, 2014.

VI S. M. Muddala and M. Sjöström and R. Olsson. Virtual View Synthesis Using Layered Depth Image Generation and Depth-Based Inpainting. In *Manuscript*, 2015.

Terminology

Abbreviations and Acronyms

1D	One-Dimensional
2D	Two-Dimensional
2DV	Two-Dimensional Video
3D	Three-Dimensional
3DTV	Three-Dimensional Television
3DV	Three-Dimensional Video
ATTEST	Advanced Three-Dimensional Television System Technologies (EU project)
CDD	Curvature Driven Diffusion
DIBR	Depth-Image-Based Rendering
DES	Depth Enhanced Stereo
DM	Disocclusion Map
FTV	Freeview point Television
HDTV	High Definition Television
HVS	Human Visual System
IBR	Image Based Rendering
IST	Information Society Technologies
ITU	International Telecommunication Union
LCD	Liquid-Crystal Display
LDI	Layered Depth Image
MPEG	Motion Picture Expert Group
MRF	Markov Random Field
MSSIM	Mean Structural Similarity Index Metric
MVC	Multiview Video Coding
MVD	Multiview Video plus Depth
MVP	MPEG-2 Multiview Profile
PC	Pair Comparison
PDE	Partial Differential Equations
YPSNR	Peak Signal-to-Noise Ratio for luminance channel
RGB	Red, Green, Blue

V+D	Video plus Depth
VSRS	View Synthesis Reference Software
YCbCr	Color space with one luma component and two chrominance components, blue and red

Mathematical Notation

α	Scalar value
β	Depth enhancement parameter
χ	Depth tolerance threshold
ϵ	Scalar constant
γ	Depth variance threshold
λ	Interpolation parameter
μ	Percentage of depth threshold
σ	Standard deviation
ξ	Threshold to identify the FG objects pixels
$\nabla\cdot$	Divergence operator
Δ	Laplacian
B	Baseline
C	Confidence term
D	Data term
D_{diff}	Diffusion term
F	Fundamental matrix
E	Convolution kernel
G	2D Structure tensor
J	3D Structure tensor
$\hat{L}$	Depth regularity term
N_{b}	Number of best patches
P	Priority term
Y	Pixel value in depth map
T_{U}	Upper depth threshold
T_{L}	Lower depth threshold
Z_{near}	Minimum depth value
Z_{far}	Maximum depth value
d	Disparity
f	Focal length
g	Control function
k	Scalar curvature
w	Weighting coefficient
x	X-coordinate
y	Y-coordinate
z	Z-coordinate

z_O	Original camera depth value
z_V	Virtual camera depth value
$\mathbf{p}, \mathbf{q}, \mathbf{r}$	Pixels in an image
$\mathbf{m}$	Camera pixels coordinate
$\mathbf{m}_\mathrm{O}$	Original camera pixels coordinate
$\mathbf{m}_\mathrm{V}$	Virtual camera pixels coordinate
$\mathbf{C}$	Camera center
$\mathbf{I_d}$	Identity matrix
$\mathbf{K}$	Intrinsic parameters matrix of a camera
$\mathbf{P}$	Projection matrix of a camera
$\mathbf{R}$	Rotation matrix
$\mathbf{P}_\mathrm{O}$	Projection matrix for original camera
$\mathbf{P}_\mathrm{V}$	Projection matrix for virtual camera
$\mathbf{t}$	Translational vector
$\mathbf{t}_\mathrm{L}$	Translational vector for Left view
$\mathbf{t}_\mathrm{R}$	Translational vector for Right view
$\mathbf{t}_\mathrm{V}$	Translational vector for Virtual view
$\mathbf{I}$	Image or video frame
$\mathbf{H}$	Hole mask image
$\mathbf{Z}$	Depth image
$\mathbf{Z}_\mathrm{A}$	Average depth threshold image
$\mathbf{Z}_\mathrm{B}$	Background depth threshold image
$\mathbf{Z_p}$	Depth patch centered at pixel $\mathbf{p}$
$\nabla\mathbf{I}$	Image gradient
$\partial\mathbf{I}$	Partial derivative of I
$\mathbf{I_p}$	Image intensity value at pixel $\mathbf{p}$
$\mathbf{M}$	3D World Coordinate
Ω	Empty region or Hole region
Φ	Source region
Φ_B	Background source region
Φ_F	Foreground source region
$\delta\Omega$	Boundary region
$\delta\Omega_1$	One-sided boundary region
$\delta\Omega_1^{'}$	Depth-guided boundary region
$\delta\Omega_\mathrm{F}$	Foreground boundary region
$\delta\Omega_\mathrm{B}$	Background boundary region
$\Psi_\mathbf{p}$	Patch centered at pixel $\mathbf{p}$
$\Psi_{\hat{\mathbf{q}}}$	Estimated source patch centered at $\hat{\mathbf{q}}$ from patch matching

Chapter 1

Introduction

Three-dimensional (3D) technology is developing rapidly with a wide range of applications such as robotics, metrology and medicine. One of the most popular applications of the 3D technology in visual communications is three-dimensional video (3DV). The trend in 3DV, which uses stereoscopic technology continues to grow with the success of the cinema and television industry. However, developing this technology further in order to provide an immersive experience is challenging.

The challenges are spread across various stages of 3DV capturing, distribution and rendering, which are vital steps for producing high quality 3D contents for the viewers. This dissertation mainly addresses the problems associated with the rendering stage in order to facilitate high quality content generation for 3DV.

1.1 Motivation

In recent years, 2D Video has experienced significant technological development including advancements in quality, larger displays and new, interactive environments. In addition, the focus in this area is now shifting from 2D to 3D Video. Adding depth as a new feature to 2D Video and providing a natural feeling of the real world, 3D video has become increasingly popular [Onu11]. Moreover, the commercial success of 3D movies in the cinema and advancements in 3D technologies has increased the interest towards developing 3D products for home and office environments. The applications for 3DVideo are 3D TV, FTV and immersive video conferencing. These technologies aim at providing an immersive 3D experience and look around possibilities to navigate around the scene in living room and office environments [Tan06]. 3D TV that can be experienced without additional eye-wear is expected to be the next generation of home entertainment after HDTV. Traditional stereoscopic 3DTVs require special glasses to perceive the 3D feeling. Wearing special glasses in movie theatres has been acceptable, but in the home environment it is considered cumbersome. Multiview auto-stereoscopic displays (or in short: multiview displays) are currently available in the market with developments in display technology [Phi, MVi]. Multiview displays provide multiple perspective of the same scene and enable a multiuser sensation to see 3D and look around the scene without any additional eye-wear. Therefore it is possible to envision that viewers can experience immersive 3D visual content without requiring any special glasses in the home or in office environments.

To provide an immersive 3D experience, displays require a number of video captures from different perspectives (a.k.a. viewpoints). However, it is not a feasible solution to capture and transmit a large number of viewpoints required by the display. As a solution, specific data conversions and rendering techniques can be employed to generate suitable content for these applications [MSD$^+$08]. Therefore, rendering plays a key role in producing content for 3D displays. One exemplary scenario for producing multiple views for a multiview 3D display is shown in Fig. 1.1. The 3DV format in Fig. 1.1 is Multiview Video plus Depth (MVD), which uses multiple regular 2D videos and an additional scene geometry (a.k.a. depth) in the form of an 8-bit gray scale sequence [SMM$^+$09b]. In this dissertation, first the problems associated with the depth-image-based rendering method (DIBR) are analyzed. Then using that knowledge, new rendering methods are introduced for different camera setups.

1.2 Purpose

Rendering plays a significant role in the content generation process by producing the desired number of views in order to create multi-scopic 3D presentation. The purpose of this research is to investigate problems associated with rendering virtual views and to facilitate high quality content generation in order to improve the quality of the 3D video experience.

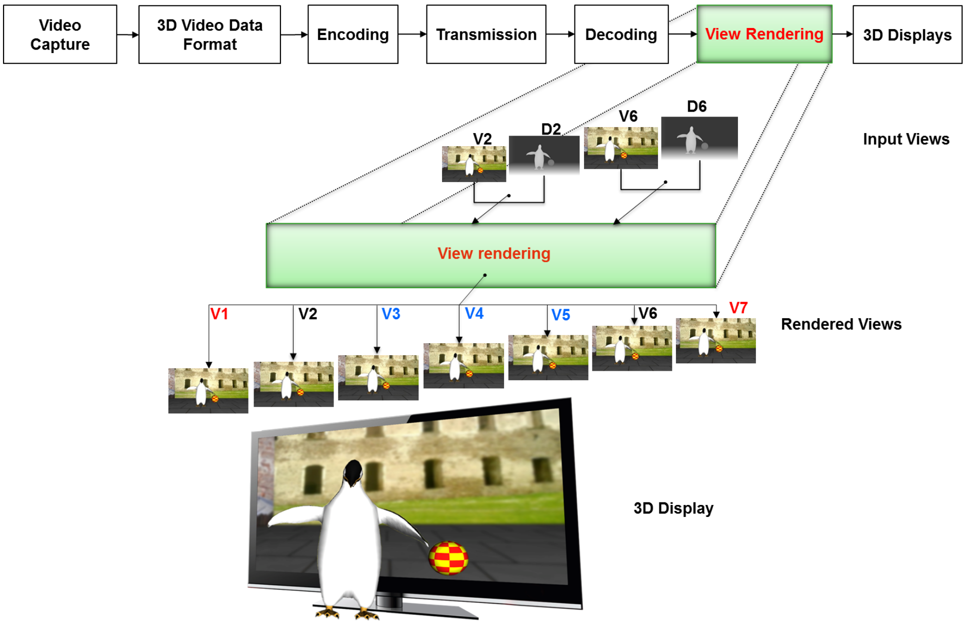


Figure 1.1: The view rendering process for a multiview display using a set of views plus depth information (texture images are represented with V and depth images are represented with D, followed by a number indicating the view point).

1.3 Problem definition

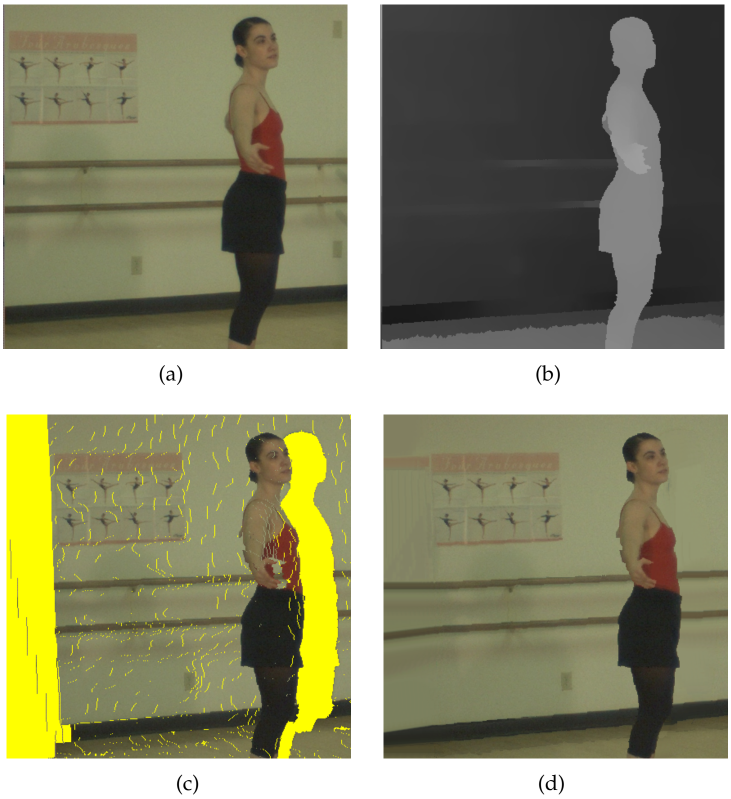
Generating 3D video of high quality is a challenging task, in general. In the context of multiview displays and free-view selection, there are many issues, especially when a limited number of views are available. An example of rendered images using DIBR methods from texture and depth (V+D) are shown in Fig. 1.2. Existing methods for rendering virtual views face the following problems:

1. Rendered views still exhibit artifacts, consequently 3D video quality is reduced.

2. Hole-filling in the rendered views in a visually plausible manner is still a challenge.

3. Spatial inconsistency in the virtual view reduces the 3D video quality.

Understanding the causes of rendering problems and providing consistent solutions can improve the 3D video quality.



(a) (b)

(c) (d)

Figure 1.2: View synthesis results for an image of the sequence "Ballet": (a) Texture image; (b) Depth image; (c) Rendered view before processing (holes in yellow color); (d) Rendered view after processing using methods presented in this thesis.

1.4 Objectives

The research work presented in this dissertation aims to improve the 3D video quality experience by improving the rendering process for stereoscopic and multiview auto-stereoscopic displays. The approach is to understand the reasons behind the rendering problems and then to provide consistent solutions. This approach is the basis for the following list of objectives:

O1 To investigate rendering artifacts in the DIBR method.

O2 To propose an alternative rendering solution that reduces artifacts.

O3 To investigate and analyze perceived visual quality of the rendered virtual views.

O4 To investigate hole-filling methods in the rendering process and to propose a depth-based inpainting method to reduce artifacts.

O5 To investigate the influence of depth information at various steps of the inpainting process and to propose a solution that address depth-based inpainting artifacts.

O6 To investigate a spatial and view consistent rendering solution that avoids disocclusions in the rendered view.

1.5 Scope

The work in this dissertation falls within the signal and image processing field, and addresses one of the key problems in the 3D Video distribution chain: rendering. The general concept is illustrated in Fig. 1.1. This dissertation aims for obtaining the best possible quality from the available data in 3D images in the context of rendering.

3D images based on given 2D images with supplemented depth information are used. Both single given view i.e. Video plus Depth (V+D), and multiple given views i.e. Multivew Video-plus-Depth (MVD) formats are considered [DGK$^+$09, SS02]. As the main focus is on the artifacts caused by the DIBR process, other artifacts such as the ones due to compression of the 3D images are not considered in the thesis.

This dissertation is focused on achieving the best possible quality for the unknown areas, a.k.a. holes, in the virtual views. Information from the given 2D image and supplemented depth are the sole sources of information for the hole-filling process.

The rendering methods in this dissertation allow for small and large displacements of the virtual camera. They also allow for horizontal and arc displacement, as well as free displacement of the virtual camera. A horizontal displacement of the virtual camera is used in Chapter 6 in relation to the interpolation based hole-filling method, and both the horizontal and arc displacement of the virtual camera are used in the parts about the inpainting based method for hole-filling.

The work focuses on applications where the virtual views will be presented on standard 2D displays (e.g. FTV), on stereoscopic displays, and on multiview autostereoscopic displays.

1.6 Thesis outline

The following chapters in this dissertation are organized as follows: Chapter 2 will provide an overview of 3D video. It will cover human depth perception basics, 3D evolution over the years, the general concept and parts of distribution chain. In Chapter 3, an overview of rendering methods and the principles behind the DIBR method are presented. In continuation of the theoretical basis related to this dissertation, the overview of inpainting methods and principles are presented in Chapter 4. The related work in the context of rendering methods and inpainting are presented in Chapter 5. The proposed edge-aided rendering method for V+D and MVD are presented in Chapter 6 together with the visual quality of methods and analysis of rendering artifacts. Depth-based inpainting methods for addressing holes in the rendered views are presented and the various parts in the inpainting method are analyzed in Chapter 7. Based on the understandings of rendering methods and inpainting methods, a layered depth image-based rendering is presented in Chapter 8. Finally, in Chapter 9 the dissertation is concluded and future work is discussed.

1.7 Contributions

The author's contributions to this are presented in five published papers (four conference papers and one journal paper) and one manuscript. The co-authors have contributed to the formulation of the methodology and the analysis. Thus the word "we" in this dissertation refers to the author and co-authors. The author of this thesis has contributed to the ideas for papers II to VI, the implementations and selections of evaluation methodology, and the analysis for papers I to VI. The contributions of this dissertation are:

PI Proposing an alternative DIBR solution with the idea of introducing edge-pixels,

PII Presenting and evaluating an extension of the edge-aided rendering method in Paper I for multiview video plus depth data with subjective evaluations.

PIII Proposing and evaluating a method to fill the holes in the rendered views using depth-based inpainting.

PIV Analysis of depth influence in various stages of the inpainting method (a.k.a. texture synthesis), presented in Paper III.

PV Presenting a depth edge-based source region classification in the inpainting process.

PVI Presenting and analyzing of an alternative formulation for rendering using layered depth generation and inpainting.

The above contributions in terms of ideas, formulations, implementations and evaluations are detailed in Chapters 6 to 8. The dissertation is structured as follows: Paper I and II are presented in Chapter 6, Paper III to V are presented in Chapter 7, and finally the manuscript, Paper VI, is presented in Chapter 8.

Chapter 2

Three Dimensional Video

This chapter provides a general background on depth perception, as well as on 3D video and its history, including a brief description of relevant parts of the 3DV distribution chain. A more detailed discussion on the presented topics is outside the scope of this dissertation, interested readers are referred to the relevant literature on each topic and other publications [JO02, SKS05, FWE13].

2.1 Human depth perception

Depth is defined as the relative distance between the viewer and an object in the 3D world. The Human Visual system (HVS) perceives the depth by sensing the relative distance and combining data in the brain. The idea behind the 3D technology is to provide consistent depth cues (visual information) to the human visual system, just as if it sees the 3D world through the naked eye. Therefore, the 3D technology creates an illusion through providing consistent depth cues in order to perceive the depth through 2D images. For this purpose, the knowledge about the characteristics of HVS and how this illusion is created is crucial to be able to perceive the depth using the display and the 3D technology.

HVS consists of two eyes, horizontally separated by an average of 65mm, and the brain to process visual signals. Each eye receives an image with a slightly different perspective compared to the image received by the other eye, then the human brain fuses the image pair to perceive the depth. HVS uses a variety of cues in the process of providing the depth perception. The visual cues can be divided into monocular cues and binocular cues. The depth sensation observed through a single eye is referred to as monocular cues. Examples include 2D image or video in traditional 2D TV. Binocular cues require two eyes to provide depth sensation. All the monocular and binocular cues should be as consistent as possible in order to ensure a comfortable viewing experience [CV95]. Important depth cues in the two classes are given as follows:

2.1.1 Monocular cues

Accommodation: It is the ability to change the focus of the lens in the eye. The ciliary muscles around the lens of each eye control the focus of the lens. The human brain uses the change in the focus and provides information about absolute or relative depth. For instance a thin lens focuses on far objects, whereas a thick lens focuses on near objects.

Motion parallax: Objects presented at different distances with respect to the viewer move with different velocities when the viewer changes position. One example of this is that farther objects move slower than near objects when the viewer is looking out from a moving train.

Occlusion: Overlapping of objects conveys the information about the order or the relative depth of the objects. Occlusion cue is also known as interposition. Example of a case: farther objects disappear when the near objects overlap them.

Perspective: Parallel lines appear to meet at a vanishing point on the horizon line. The vanishing points give information about the farthest points. One example is looking at a rail track. Perspective cues give the relative distance when the objects are known to be the same size but the absolute depths are unknown.

Shadows: The reflected light and shadows from a given surface give information about the shape and the relative depth of the objects.

Size: The varying size of an object gives information about its depth. In fact, objects closer to the eyes occupy a large visual angle on the retina than farther objects. Moreover, familiarity with the size of the objects gives information about their depth.

2.1.2 Binocular cues

Stereopsis (binocular disparity): As the eyes are separated horizontally by some distance, the projected points of the 3D world will be at different positions on the left and right eyes' retina respectively. This difference is referred to as disparity, which is a necessary depth cue in order to perceive depth.

Convergence: It is another binocular depth cue, extracting the depth information from the location of the object. When an object is in focus, two eyes are fixated on the object using extra ocular muscles. The angle between the eyes provides the depth cue. Larger angles correspond to farther objects, while for the closer objects, the convergence angle will be small.

A detailed description of stereopsis is presented here because of its importance to provide depth information in 3D Video. In the following explanation, human eyes are replaced by the horizontally separated cameras. The set-up is shown in Fig. 2.1. It is visible that the object is slightly shifted in the left and right images. The combined left and right camera images are called the stereo image. Two corresponding points are shown in the stereo image. By definition, the disparity is the difference of these corresponding projected points. An important observation from the stereo image is that the disparity of the farthest object "the tree" is less compared to the closest object "the rabbit" which has the highest disparity for parallel cameras or eyes (see Fig. 2.1).

Stereo displays use this principle to create depth impressions. When the display sends each view to the respective eye the viewer will perceive the depth using the binocular disparities in the two images. Moreover, by adjusting the disparity, the object will be placed at different depth levels in the space: (i) Objects will be placed behind the display screen when there is positive disparity, (ii) Objects will be positioned on the display screen when there is no disparity (the case of 2D), (iii) Objects will be in front of the display screen when there is negative disparity (see Fig. 2.2).

3D videos on stereoscopic displays may introduce an accommodation and convergence problem, where the viewer's eyes are focused on the display screen, but the binocular disparity provides depth at different levels. This conflict may cause visual strain [SMM$^+$09a]. The detailed discussion about depth cues and their influence is outside the scope of this dissertation. Interested readers are referred to the detailed discussion on the effect of depth cues in [ISM05, CV95].

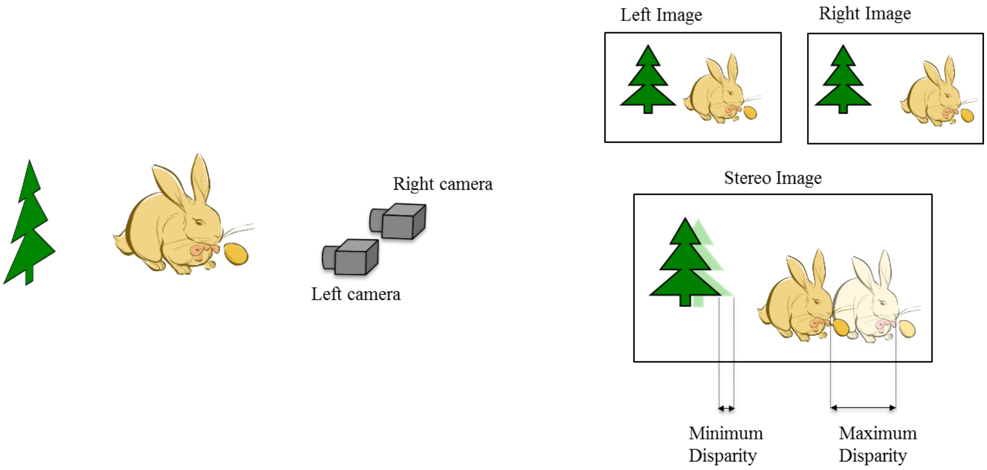


Figure 2.1: Stereopsis.

2.2 History of 3D

The history of 3D technology is traced back to 1838 when Charles Wheatstone invented a stereoscopic device to create the illusion of depth by means of the principle of binocular vision [Whe38]. The term “3D” comes from an extra dimension, which is depth, which we perceive through the stereo image. The stereo image is a pair of images of a scene with different perspectives. The system utilized to show stereo images is called the stereo system. In general, we see stereo images but we perceive depth with our brain, commonly called the 3D effect. The principle in the Wheatstone stereoscope is to show different perspective images to each eye, which creates the illusion of depth. Later, influenced by the stereoscopic filed. A new stereoscope device was developed by Sir David Brewster, who had been influenced by the stereoscopic field, in 1844. Since then many technologies have emerged. In 1903, the Lumière brothers showed the first short 3D motion picture to the public; these screenings were only watched by one viewer at a time. With the stereoscopic cinema boost in the 1920s, the first 3D feature film was released in 1922 using the anaglyph (filtering by complementary colors, e.g. red and green) eye-wear. Using the principle of the stereoscopy, the first experiments on 3DTV were conducted in 1928 [Til11]. Despite these early successful experiments of the stereo cinema and television, it took nearly two decades for Hollywood to tune into 3D movies. With the tremendous success of 3D movies, Hollywood produced over sixty five 3D movies between 1952 and 1954. Later, the early success of 3D movies was hampered by insufficient technology, inadequate quality control and lack of stereographic experience. With the development of technology, solid understanding of the depth perception and quality control over the years, 3D cinema was revived with the movie “Avatar” in 2009. Since then the production of 3D movies is growing.

Similar to the 3D cinema, television also had difficult phases to tune into 3D tech-

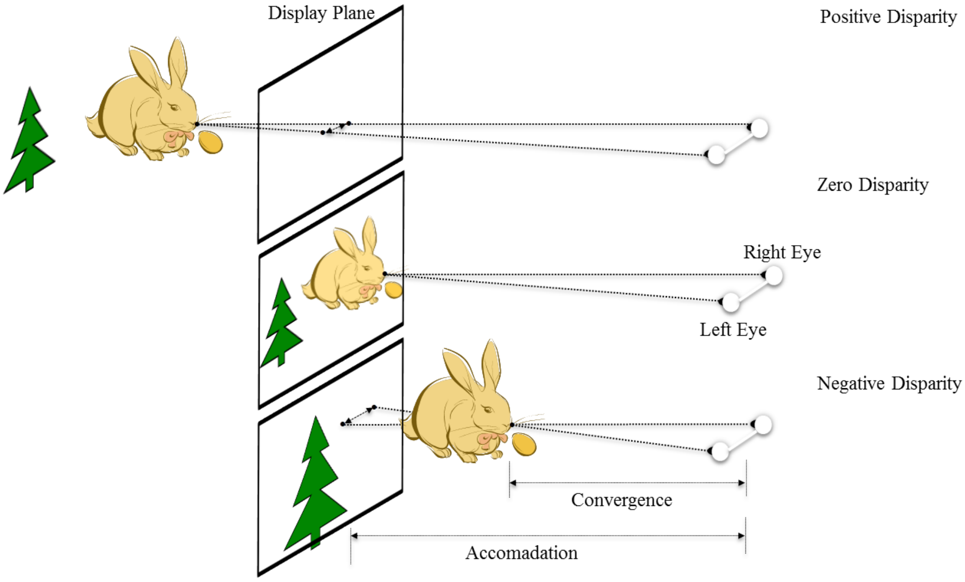


Figure 2.2: Binocular disparity.

nology. Despite the difficulties in transmission and displays, the first non-experimental stereoscopic broadcast was aired in 1980. However, the limitations in the analogue TV and gradual transition from analogue to digital service gave rise to 3DTV research efforts in the early 1990s'. Later, inspired by the interest in 3DTV broadcast services and to provide backward compatibility for conventional 2DTV, the Motion Picture Expert Group (MPEG) started working on a compression technology for stereoscopic video sequences that resulted in a Multi View Profile (MVP) as part of the MPEG-2 standard [IL02].

Stereoscopic broadcast started with the Winter Olympics in Nagano in 1998, after integrating stereoscopic TV and HDTV into the high quality 3D entertainment medium [JO02]. The first stereoscopic channel was started in 2010 in South Korea. Over the years, the 3D viewing experience was greatly improved by means of advanced digital cameras and a deeper understanding of the depth perception. Moreover, the development of digital technology and 3D displays improved significantly, which created more interest in MultiView Video (MVV) applications. Improvements in 3D video technologies raised more interest in 3DTV and FTV [MFW08]. 3DTV provides depth impression without aided wear, whereas FTV allows the user to choose the viewpoints. A more comprehensive history of the 3D technology is provided in [FWE13].

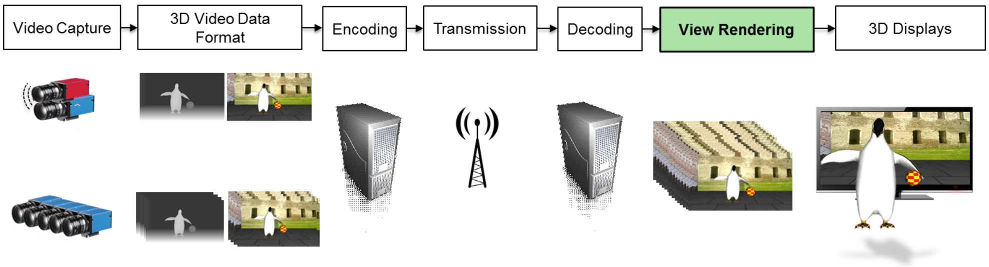


Figure 2.3: Typical 3DTV and FTV broadcasting chain.

2.3 3D video transmission chain

The promising 3D video applications include 3DTV and FTV. They create the illusion of depth and navigation in the scene. 3DTV provides depth experience. The depth experience is created by the display, which projects a separate view for each eye. Although two views are sufficient to provide the impression of depth, higher number of views improve comfort and the depth perception. FTV offers the user to select a desired view point, which creates the illusion of navigation in the 3D scene [Tan06]. To be able to provide these experiences, both the technologies 3DTV and FTV require a number of views. A typical transmission chain to distribute 3D Video is shown in Fig. 2.3.

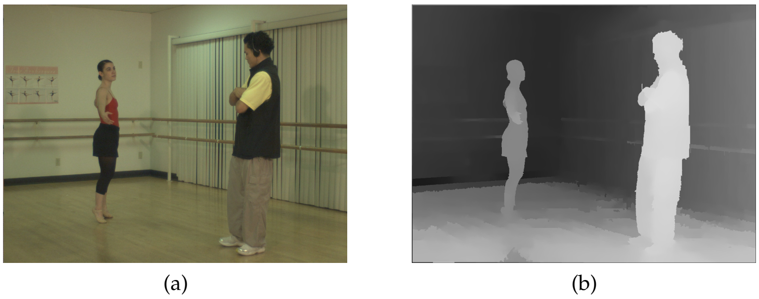
Traditional 3DTV relies on the concept of stereoscopic video, where two views are captured, transmitted and displayed using stereo displays. Later, capturing and bandwidth limitations were introduced to the transmission chain. A 3DTV framework was presented by the European Information Society Technologies (IST) project "Advanced Three-Dimensional Television System Technologies" (ATTEST) [RMOdBaI$^+$02]. Subsequently, 3D4YOU, Multimedia Scalable 3D for Europe (MUSCADE) and more advanced projects were added with varying 3D video formats aimed at supporting the wide range of 3D displays and applications [BVG$^+$11], [Mus]. The key steps in the video framework consist of: representing the captured data with post-processing, compressing the data efficiently, and finally, depending on the application, rendering the required views.

2.4 3D video representation

3D video representation (a.k.a. 3DV-format) is a process of converting captured video data into a flexible video format in a way that the converted data allows efficient compression and effective rendering. The formatted data are encoded and transmitted through channels (see Fig. 2.3). However, 3D video representations have different advantages and disadvantages over conventional stereo video representation. The advantages of the 3DV-format include reduction in the bandwidth consumption, backward compatibility with 2DTV and depth adjustment at the receiver. Despite the advantages, several disadvantages are also associated with the 3DV-formats, which must be considered in order to produce high quality 3D. It is worth noting that the quality of the rendered view depends great extent on the quality of the input data and the rendering. Thus, creation of the input data and rendering become extra important. Although there are several types of 3DV formats, the following formats are commonly used in the distribution of 3D video: video plus depth, multiview video plus depth and layered depth video [SMM$^+$09b].

2.4.1 Video-plus-depth

Video-plus-Depth (V+D) is one of the efficient data formats introduced for 3D video instead of the stereo video. The data in V+D consist of a regular 2D video and the depth per pixel information (see Fig. 2.4.1). The 2D video provides the information about the color, texture and structure of a scene, whereas the depth provides the information about the distance to the 3D scene for each pixel from the camera center. The real depth values are translated into image intensities; therefore the depth image often represented as a 8-bit gray-scale image. Brighter values in the depth image represent nearer points to the camera, whereas darker points correspond to farther points. At the receiver side, the depth values are translated back to the original depth values in order to render additional views. The European project ATTEST has given that the depth data can be compressed to about 20% of the overall bit rate. Although the 3D data are efficiently transferred to the end user with the V+D format, a major problem associated with this data format is disocclusions in the rendered views. Chapters 6, 7 and 8 address the disocclusion problem in different ways.


(a)

(b)

Figure 2.4: 3D Video format: Video+Depth: (a) Texture image; (b) Depth image.

2.4.2 Multiview video-plus-depth

The Multiview-Video-plus-Depth (MVD) format is introduced to avoid problems associated with the V+D format and to be able to provide immersive depth by producing in between views. The MVD format is an extension of V+D, it consists of two or more V+D data (see Fig. 2.5). Using MVD data, the views in between the V+D data can be effectively rendered, since the texture for disocclusions can be found in other views in MVD. However, when the display requires many views to create a smooth transition between views, it still has to use the outermost V+D in MVD to extrapolate virtual views. Therefore MVD has the same disadvantage as the format based on a single V+D.

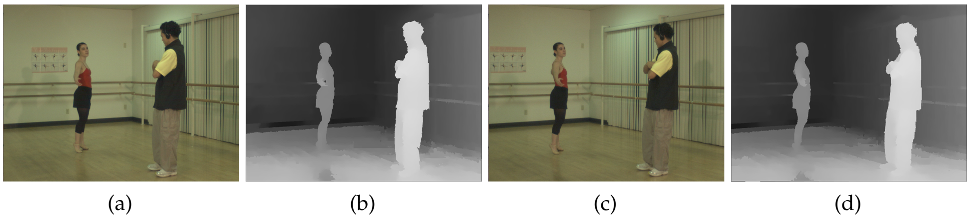


Figure 2.5: 3D Video format: Multi-view video plus depth (MVD): (a) Texture image view 3; (b) Depth image view 3; (c) Texture image view 5; (d) Depth image view 5.

2.4.3 Layered depth video

Layered depth image concept was first presented in [SGHS98] for complex geometries. Unlike the 2D image, the LDI contains multiple pixels at different depth layers. Each depth layer consists of a set of pixels. The layers are arranged from the front layer to the back. The front layer corresponds to the nearest scene to the camera (also called the main layer) and the next layer corresponds to farther (hidden) objects. By definition, the number of layers is not limited. However, in practice, only a limited set of layers are used. Layered Depth Video (LDV) is a temporal extension of the LDI format. The limitation associated with the layered depth formats is to capture the hidden layers. Alternatively, the LDV can be generated with MVD by using the warping technique. Therefore it is called a derivative of MVD. An example illustrating the LDI format is shown in Fig. 2.6. The main layer in the LDV format corresponds to the V+D. The next layers correspond to hidden texture and depth information. The extension of LDV is depth enhanced stereo (DES) consists of conventional stereo videos with LDI. A more detailed description of different 3DV formats is given in [SMM$^+$09b].

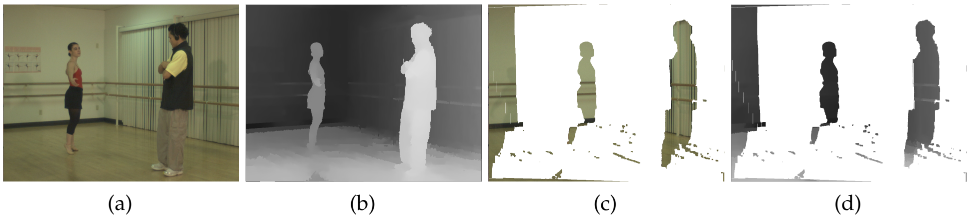


Figure 2.6: 3D Video format: Layered Depth Image (LDI): (a) Texture image; (b) Depth image; (c) Layered texture image; (d) Layered depth image.

Disocclusions in the rendered view are expected to be filled using MVD or subsequent formats (LDI, DES); however, the amount of the disocclusion to be filled depends on the viewing angle of the captured MVD. Therefore, disocclusion handling might still be required depending on the display requirements. The 3DV formats V+D, MVD are used in this dissertation to render additional views.

2.5 Rendering

Once the transmitted data have arrived at the receiver side, they are decoded into the color video and the depth data. Using these data, a number of additional views are generated using the DIBR algorithms. These generated views are then presented on stereoscopic or multiview auto-stereoscopic displays, depending on the application. The auto-stereoscopic displays and FTVs require many views to provide immersive depth impression. Producing these many views with a high quality is challenging. As this thesis addresses the rendering problems to improve the rendered view quality, more details about DIBR are presented in the subsequent Chapter 3.

2.6 3D displays

The key idea behind 3DV technologies is to create depth and look-around impressions, and so they require displays, which offer such facilities. Over the years, many displays were produced to provide such an experience. The importance of stereo image and disparity, which create the depth impression was mentioned in Section 2.1. Based on the knowledge about the HVS, the stereoscopic 3D displays have been implemented providing proper visual information to each eye. All the stereoscopic displays require at least two views of the same scene captured from different perspectives.

The 3D display technology can be broadly categorized into stereoscopic displays and auto-stereoscopic displays based on the need for supporting eye-wear to perceive 3D. Stereoscopic displays require the viewer to wear some kind of additional device to direct the left and right views into appropriate eye. In contrast, auto-stereoscopic displays do not require any additional devices since necessary optical elements are directly integrated in the displays. Employing more number of views to see different perspective from different viewpoints.

2.6.1 Stereoscopic displays

The stereoscopic displays provide left and right views to the corresponding eye by using different types of multiplexing methods. These displays can be categorized by the type of multiplexing technique used, namely: color multiplexing, polarization multiplexing, and time multiplexing displays. Color multiplexing, also known as anaglyph displays, combine the left and right images using the near complimentary

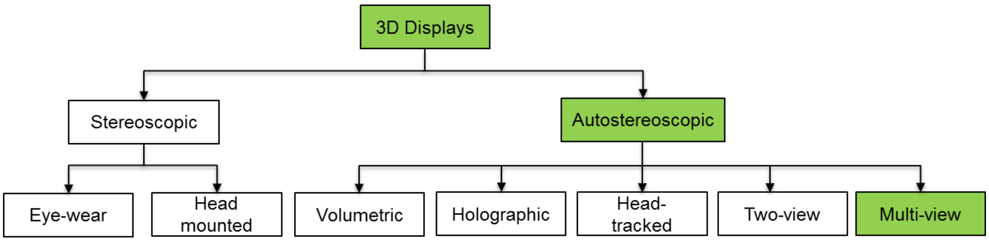


Figure 2.7: Taxonomy of 3D displays.

colors (red and green, red and cyan or green and magenta). Viewers wear a pair of anaglyph glasses to separate views and thus receive corresponding images for the left and right eye. However, these displays have serious problems with loss of color information and crosstalk. The problems are minimized by using wavelength multiplexing with dichromatic interference filters [JF06]. 3D cinema mostly uses polarization multiplexing techniques, in which two views with different perspectives are projected onto a display using orthogonally polarizing filter sheets (linear or circular polarization) [Pas05]. Viewer use polarized glasses to perceive 3D, since polarized glasses allow only one view to pass each eye. Time-multiplexed displays use the memory effects of the HVS and display left and right views alternatively. These displays require active shutter glasses, which are synchronized with the display. Another type of stereoscopic display is a head-mount display, where the viewer wears the device instead of the eyewear. The device consists of two displays, which direct the left and right images to the respective eye. The viewer perceives 3D since the viewer sees two different perspectives of the same scene [Pas05].

2.6.2 Autostereoscopic displays

Unlike stereoscopic displays, auto-stereoscopic displays do not require additional wear to perceive 3D, since the display itself separates left and right views automatically. They use a parallax barrier or lenticular lenses to provide different views for the left and right eye as shown in Fig. 2.8. Moreover, they create a look-around and motion parallax.

Two view displays provide the viewer with a stereo pair, one image for each eye, which can be achieved by multiplexing views or by two LCDs. By repeating the views a motion parallax can be created. These displays can be classified according to their optical components that direct the views to the viewer. They are parallax barrier displays and lenticular displays. In parallax barrier displays, a strip of black mask is placed to block the light. These masks are placed either in front or behind the LCD displays, which direct half of the pixels to one eye and other half to the other eye. These parallax barrier displays only provide a horizontal parallax. However, the drawbacks with these displays are loss of brightness and lower spatial resolution. The development of a slanted barrier system improved the spatial resolution to some extent. However, lenticular displays use cylindrical lenslets with flat panel

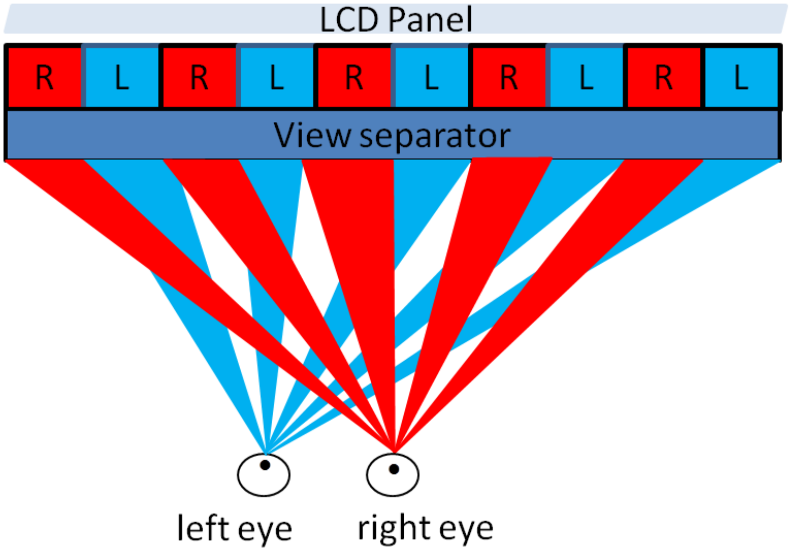


Figure 2.8: Illustration of auto stereoscopic display view generation.

displays to direct the light in certain viewing angles. Lenticular displays have problems with lenticular sheet alignment and intensity variations. Associated problems can be partially addressed by using slanted lenticular arrays.

Multiview displays provide a large number of stereo pairs which create a look-around feeling. Moreover, multiple viewers can use the display simultaneously from different locations. Less numbers of views result in jumps between the views when moving the head in front of the display. To get a smooth motion parallax when the viewer is changing his position, a higher numbers of views are required. Lenticular displays are one of the most common multiview auto stereoscopic displays. They use slanted lenslets to reduce the low resolution problem [LR06].

Head tracked displays are similar to the two-view displays with a head tracking feature, which tracks the head movement and provides the appropriate views to the eyes. Modern 3D displays like holographic, light filed, and volumetric displays can solve the accommodation convergence problem but still are in the research phase. Comprehensive summaries on the 3D displays are given in [Pas05, UCES11].

2.7 Depth acquisition

Depth information plays a critical role in the transmission and rendering for generating new views from the existing views. By definition, the depth map represents the geometry of the scene, the distance between the camera and objects in the 3D world. The depth information can be generated by using computer graphics, range sensors and stereo analysis [SOS13]. Both computer generated depth and depth from stereo matching are used to render additional new views in this dissertation. Computer generated depth provides the true distances but it is not real data. However, computer generated depth avoids the problems of color differences and inconsistencies on smooth regions, which appear with the depth from stereo analysis. Another type is the depth extracted from range sensors, either triangulation sensors or time-of-flight based sensors. The main principles of these sensors are triangulation and time travel between sender and receiver signals. A major problem associated with range sensors is the resolution mismatch, i.e. the resolution of the depth image is smaller than the resolution of the texture image. This leads to severe artifacts in the rendered views. Depth image up scaling is a method to produce the required resolution of depth image as texture image resolution. One of the efficient depth image up scaling method was introduced in [SSO14].

One of the common methods to extract the depth is by using stereo matching, i.e. estimating the disparities between the views. The relation between the disparity d and the depth z is given as:

$$z = \frac{Bf}{d}. \tag{2.1}$$

where f is the focal length and B is the distance between stereo cameras. The disparity is estimated from the corresponding pixels between the two views on the epipolar line and this process is referred as disparity estimation. The definitions of corresponding points and the epipolar line are given in Chapter 3 in details. Once the disparity for each pixel is known, the depth value can be determined using maximum and minimum disparity as follows:

$$Y = 255 \cdot \frac{d - d_{min}}{d_{max} - d_{min}}. \tag{2.2}$$

Where Y is an 8 bit gray scale value, i.e. gray value 0 indicates the farthest point and the value 255 indicates the nearest point. The gray scale depth values should be translated into metric values using the nearest and farthest clipping planes using (3.8) in order to generate new views. The depth obtained from stereo matching suffers from occlusions, low texture areas and repetitive textures. These shortcomings can be reduced by various post processing techniques. Note that the errors in the generated depth map cause artifacts in the rendered views. The detailed process of obtaining the depth map is not presented in this dissertation since it is outside the scope of this work. Interested readers can find more details in [SS02, HZ04, Mor09].

Chapter 3

Depth-Image-Based Rendering

The concept of 3DTV, FTV and the distribution of 3D content using V+D, LDI and MVD formats have been presented in Section. 2.4. As known from the previous chapters, view rendering is a necessary task in order to produce a sufficient number of views to be able to provide necessary effects like smooth depth impression and navigation around the scene. The term “view rendering” is also known as “view synthesis” in the field of computer vision, in the context of producing new perspectives using available data. The data consists of regular 2D videos and associated depth information as described in Section. 2.4. Methods that rely on the depth information to render views are called depth-image-based rendering (DIBR) methods.

This chapter gives an overview of the view rendering methods together with some important basics. Starting with an overview of the methods, the projective geometry is introduced, which describes the pinhole camera model. The basic concepts of the image formation are then discussed for the case of single and two views and the relation between them. Next, the DIBR techniques to produce new views using the principles of perspective projection are presented. Finally, the artifacts and challenges associated with the DIBR method are discussed.

3.1 Overview of rendering methods

A number of input views are required in order to offer a desired depth experience and look around feeling using current displays. Moreover, the input views should be adaptive to different test setups and viewing conditions. Thus view rendering is employed as an efficient solution to generate content for desired displays. The taxonomy of view rendering methods is shown in Fig. 3.1 [KES05]. View rendering methods are basically categorized into Model Based Rendering methods (MBR) and Image Based Rendering methods (IBR). Model based methods require information about scene models to generate new views. MBR methods are mostly used for computer generated imagery. IBR methods use either intrinsic data (i.e. available images) or geometric data. IBR methods are further classified based on the availability of geometric information. DIBR methods belong to the class of explicit geometry methods since they use depth maps in the view rendering process. Note that the rendering methods in this dissertation fall into the explicit geometry class as depth is used in the rendering process.

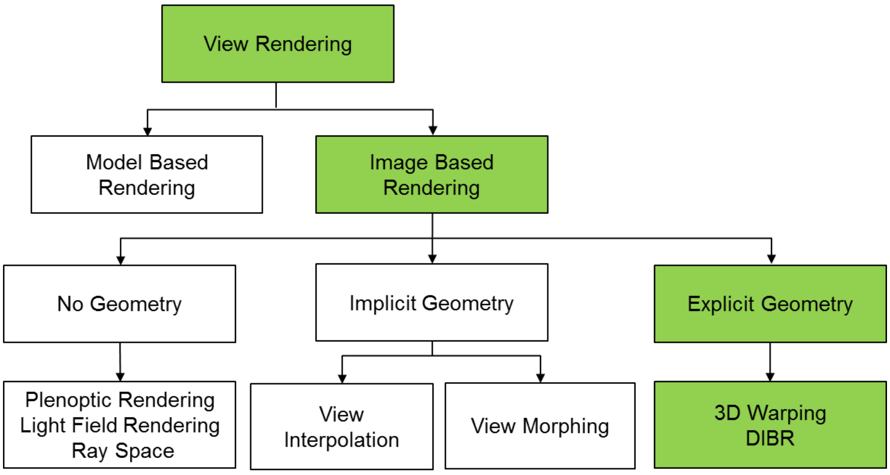


Figure 3.1: Rendering methods classification. [SK00]

3.2 Projective geometry

Projective geometry is a well-organized mathematical framework used e.g. in computer vision. The applications of projective geometry include modeling perspective projection of 3D data, scene reconstruction from multiple views, and robotic navigation. In perspective projection, two parallel lines meet at infinity at the vanishing point. Unlike Euclidean geometry, the projective geometry can model a vanishing point, which is an advantage of projective geometry. The equations in the projective geometry use homogenous coordinates, which provide a way to create (N)-D vector to $(N+1)$-D vector in projective space. For example a point in a 2D image with a 2D

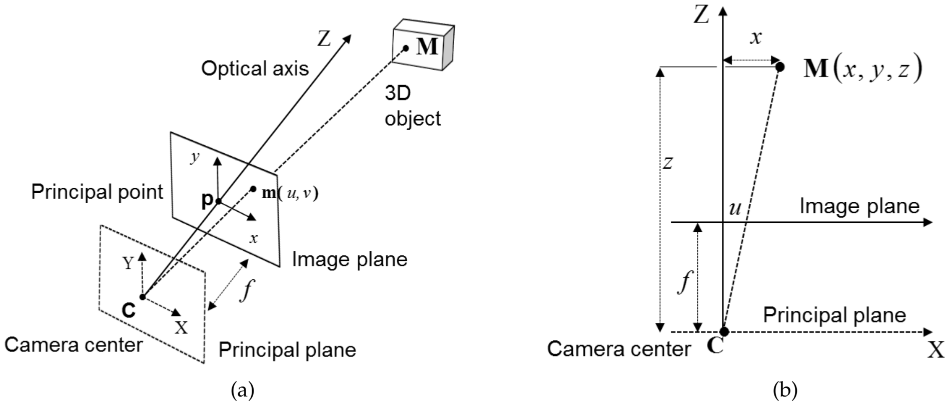


Figure 3.2: Pinhole camera model [SKS05]: (a) Projections illustration; (b) Geometric relations.

vector can be expressed by a 3D vector. As the result, equations become linear and can be expressed by using matrix operation in the perspective projection.

3.2.1 Single view geometry

The pinhole camera is the simplest camera model, which defines how a point in the 3D world is project onto the 2D image plane through a pinhole. The mapping of the points in the 3D world to the points in the 2D image is called perspective projection.

The geometry of camera model consists of an optical center (a.k.a. center of projection) and an image plane [HZ04]. The distance between the optical center and the image plane is called the focal length f and the distance to the 3D object is the depth z. The orthogonal line to the image plane passing through the camera center is the optical axis (a.k.a. principal axis) . The point where the ray and the image plane intersects is called the principle point. Moreover, the plane, which contains the camera center and is parallel to the image plane is the focal plane or the principal plane (see Fig. 3.2(a)).

Let the camera center $\mathbf{C}$ be placed at the origin of Euclidean coordinate system and that the optical axis be collinear with the z-axis. The geometrical relation between the 3D world point $\mathbf{M} = (x, y, z)^T$ and the image point $\mathbf{m} = (u, v)^T$ in the image plane is given using similar triangles (see Fig. 3.2(a)).

$$\begin{aligned} \frac{f}{z} &= \frac{u}{x} = \frac{v}{y} \\ u &= \frac{fx}{z}, v = \frac{fy}{z}, \end{aligned} \tag{3.1}$$

Expressing these relations in the homogeneous coordinate system, we get:

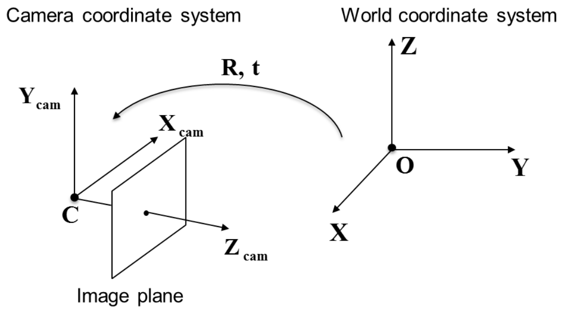

Camera coordinate system
World coordinate system
R, t
$\mathbf{Y}_{\mathbf{cam}}$
$\mathbf{X}_{\mathbf{cam}}$
C
$\mathbf{Z}_{\mathbf{cam}}$
Image plane
Z
O
Y
X

Figure 3.3: Transformation from world coordinate to camera coordinate from [HZ04].

$$z\begin{pmatrix}u\\v\\1\end{pmatrix}=\begin{bmatrix}f&0&0&0\\0&f&0&0\\0&0&1&0\end{bmatrix}\begin{pmatrix}x\\y\\z\\1\end{pmatrix}, \tag{3.2}$$

where u and v are coordinates of the pixel $\mathbf{m}$ in the image. The matrix in (3.2), which transforms the world coordinates to the image coordinates, is called the projection matrix (a.k.a. camera matrix). Expressing (3.2) in the simple form we get:

$$z\mathbf{m}=\mathbf{PM}, \tag{3.3}$$

where $\mathbf{M}=(x,y,z,1)^T$ and $\mathbf{m}=(u,v,1)^T$ are homogenous coordinates of the 3D world point and its projection point on the image plane. $\mathbf{P}$ is the projection matrix of the camera and only contains information about the focal length. However, real cameras are described by several parameters, such as position and orientation with respect to the world coordinate frame.

In general, the mapping of a 3D world point to the 2D image point is given by the transformation of the world point into camera point and then to the image point. When the camera coordinate frame is not aligned with the world coordinate frame they relate by translation and rotation. Therefore the world coordinate frame is transformed to camera coordinate frame as shown in Fig. 3.3. Then the camera center in the camera coordinate frame is expressed as $\mathbf{M}_{cam}=\mathbf{R}\cdot(\mathbf{M}-\mathbf{C})$, where $\mathbf{M}_{cam}$, $\mathbf{M}$ are the same world point in the camera coordinate and the world coordinate frame respectively, $\mathbf{C}$ is center of the camera in the world coordinate frame. After simplification and using homogeneous coordinates, the camera matrix is given as:

$$\mathbf{P}=\mathbf{K}\left[\mathbf{R}|\mathbf{t}\right], \tag{3.4}$$

where $\mathbf{K}$ is the intrinsic camera calibration matrix and $[\mathbf{R}|\mathbf{t}]$ is the extrinsic parameters matrix. The extrinsic parameters describe the camera location and orientation and specify the transformation from world coordinates to the camera coordinates system. $\mathbf{R}$ is the rotation matrix and $\mathbf{t}$ is the translation vector.

The intrinsic camera calibration matrix transforms the camera coordinates to the pixel coordinates. The intrinsic matrix describes the properties of the lenses and image sensors: focal distance f, skew parameter s, image centre o_x,o_y and camera pixel size s_x,s_y) in x and y-directions and the matrix is defined as:

$$\mathbf{K} = \begin{bmatrix} f/s_x & s & o_x \\ 0 & f/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.5}$$

In summary, the projection matrix $\mathbf{P}$ is a 3x4 full rank matrix with 11 degrees of freedom: five from the intrinsic matrix, three from the rotation matrix, and three from translation vector. The projection matrix first transforms 3D world points to the camera plane and then to the image plane. Moreover, the projection is not defined for the camera projection centre and is expressed as $\mathbf{C} = -\mathbf{R}^{-1}\mathbf{t}$. The optical ray of the image point is described as the line passing through the camera center and the point at infinity. In general the projection equation is written as:

$$z\mathbf{m} = \mathbf{PM}, \tag{3.6}$$

where z is the distance between the world point and the focal plane, also referred to as depth. Then the projected locations in the image are obtained using this projection equation.

3.2.2 Two-view geometry

The geometry of a single camera and how a 3D point is projected into the image plane is presented in Section 3.2.1. Now the geometry of two camera views (a.k.a. stereo camera setup) and how the same 3D points are related in the two cameras are briefly described. When two cameras capture the same point in the 3D scene from different perspectives, the projected image points are the corresponding points. The recovery of the corresponding points is the stereo-matching problem and this can be greatly simplified by exploiting the epipolar geometry relations.

The epipolar geometry basis is given as follows: suppose two cameras with centers at $\mathbf{C}_1$, $\mathbf{C}_2$ capture the same 3D scene from different perspectives as shown in Fig. 3.4(a). The distance between the two camera centers is called the baseline. Let $\mathbf{M}$ be an unoccluded 3D point, which is projected into two images at positions $\mathbf{m}_1$ and $\mathbf{m}_2$ respectively.

When these image points are back projected into the 3D world, they intersect at $\mathbf{M}$. All the three points $\mathbf{M}$, $\mathbf{m}_1$ and $\mathbf{m}_2$ form a plane referred to as *the epipolar plane*. The back projected ray from $\mathbf{C}_1$ to the world is imaged as a line on the image plane $\mathbf{I}_2$ and is referred to as *the epipolar line*. The camera center $\mathbf{C}_1$ and point $\mathbf{m}_1$ lie on this epipolar line in $\mathbf{I}_2$, since the epipolar line is an image of back projected ray $\mathbf{m}_1$ through $\mathbf{C}_1$. The projection of $\mathbf{C}_1$ on the image plane $\mathbf{I}_2$ is called *the epipole*. The epipolar geometry can be represented as a *Fundamental matrix*, which relates two corresponding points as: $\mathbf{m}_1^T F \mathbf{m}_2 = 0$. The fundamental matrix F is obtained from either camera matrices or corresponding points.

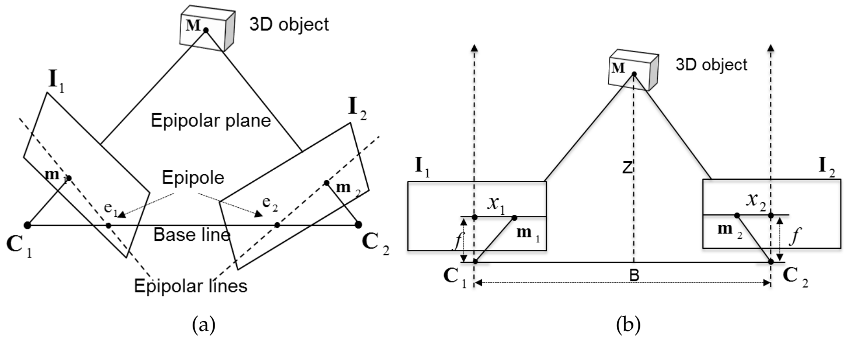


Figure 3.4: Two view geometry [HZ04, SKS05]: (a) Epipolar geometry; (b) Rectified image geometry.

Using epipolar geometry when the camera matrices are available, corresponding points can be detected by searching on the epipolar lines instead of the whole image. However, the epipolar lines are not parallel when two cameras have some rotation. In this situation, stereo image rectification is used to transfer the image planes to the same common plane. After rectification the epipolar lines will be horizontal and parallel to the base line (see Fig. 3.4(b)). Once the corresponding points are located, disparity between the stereo images (projected position difference in two perspective images of the same 3D point) can be computed. Subsequently the depth can be estimated using triangulation.

$$z = \frac{fB}{d}, \tag{3.7}$$

where z is the depth value, B is the baseline and $d = u_1 - u_2$ is the disparity along x-direction, i.e., the difference of x-coordinates of corresponding points $\mathbf{m1}$ and $\mathbf{m1}$. Using these relations, the object points in one camera view can be extracted from the other camera view.

3.3 3D warping

Depth-image-based rendering (DIBR) is a method which generates novel views using depth and camera parameters [Feh04]. In this context, we define inputs to the DIBR method as the original (a.k.a. reference) views and outputs of DIBR as virtual (a.k.a. rendered or warped) views, which consist of warped texture and depth images. The scene details can be defined as foreground and background. Foreground is the part of the scene closer to the camera that occludes other objects, and background is the part of the scene farther from the camera. The DIBR method uses texture and depth images to produce a virtual view. When DIBR takes a single V+D to produce virtual views it is called *view extrapolation*, whereas if DIBR takes two V+D to produce virtual views between the given views it is called *view interpolation*.

DIBR methods are based on the principle of the perspective projection, where the reference view is projected onto the virtual view using the depth map and camera parameters. 3D warping is a two-step process: First the reference camera image points are projected back into the 3D world. Next, the 3D world points are projected onto a virtual camera image plane.

It is known from Section2.4 that the given depth map is often represented as a 8-bit gray-scale image, which represents the depth per pixel. However, to project the image point into 3Dworld point, the gray scale depth values should be translated into metric values using the nearest and farthest clipping planes. The translation is given by

$$z = \frac{1}{\frac{Y}{255}\left(\frac{1}{Z_{near}} - \frac{1}{Z_{far}}\right) + \frac{1}{Z_{far}}}. \tag{3.8}$$

where z is a real depth value for pixel value Y in the depth image. Z_{near} and Z_{far} are depth ranges in the real scene.

3.3.1 Forward warping

The way in which 3D warping is performed is also called forward warping, where depth and texture are directly projected to virtual view. The projection of the scene into different cameras is illustrated in Fig. 3.5. Let an arbitrary 3D scene point with homogenous coordinates $\mathbf{M} = (x, y, z, 1)^T$ be projected onto a respective image plane at pixel positions $\mathbf{m}_\mathrm{L} = (u_\mathrm{L}/z_\mathrm{L}, v_\mathrm{L}/z_\mathrm{L}, 1)^T$ and $\mathbf{m}_\mathrm{V} = (u_\mathrm{V}/z_\mathrm{V}, v_\mathrm{V}/z_\mathrm{V}, 1)^T$ respectively. The projection equations are given as:

$$z_\mathrm{L}\mathrm{m}_\mathrm{L} = \mathbf{P}_\mathrm{L}\mathbf{M}, \tag{3.9}$$

$$z_\mathrm{V}\mathbf{m}_\mathrm{V} = \mathbf{P}_\mathrm{V}\mathbf{M}. \tag{3.10}$$

Re arranging (3.9) the 3D point $\mathbf{M}$ can be expressed as:

$$\mathbf{M} = z_\mathrm{L}\mathbf{P}_\mathrm{L}^{-1}\mathbf{m}_\mathrm{L}. \tag{3.11}$$

Then the relation between the virtual view pixel and the reference view pixel can be obtained by substituting (3.11) into (3.10):

$$z_\mathrm{V}\mathbf{m}_\mathrm{V} = z_\mathrm{L}\mathbf{P}_\mathrm{V}\mathbf{P}_\mathrm{L}^{-1}\mathbf{m}_\mathrm{L}. \tag{3.12}$$

Eq. (3.12) to obtain the virtual view pixel is referred to as general warping equation, in the sense that the input and virtual cameras are involved in some rotation. General camera setup causes keystone distortion in the stereo visualization that can be avoided by either capturing the content with 1D parallel camera arrangement or image rectification after capturing. In either case, the setup has only change in the x-direction. Hence, the general warping equation (3.12) is simplified into:

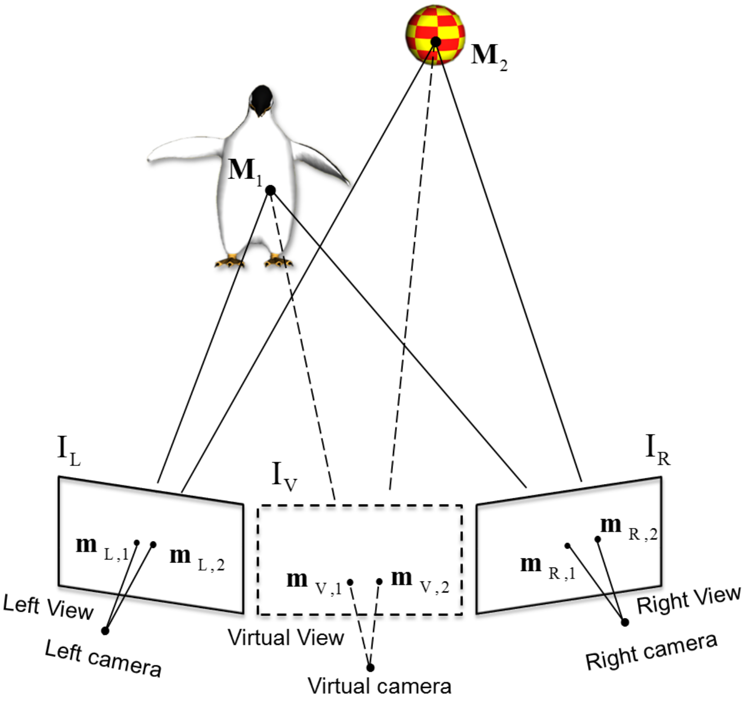


Figure 3.5: Forward warping of a pixel from left or right view into virtual view.

$$u_{\mathrm{V}} = u_{\mathrm{O}} + \frac{f \cdot (t_{x,\mathrm{V}} - t_{x,\mathrm{O}})}{z_{\mathrm{O}}} + (\mathbf{p}_{x,\mathrm{V}} - \mathbf{p}_{x,\mathrm{O}}), \tag{3.13}$$

where u is the x-coordinate of a pixel position, t_x is the x-coordinate in the translation vector, z is the real depth value and subscripts V and O represents virtual and original views respectively. $\mathbf{p}_{x,\mathrm{V}}$ and $\mathbf{p}_{x,\mathrm{O}}$ are the principal point offset for the virtual and original views. $t_{x,\mathrm{V}} - t_{x,\mathrm{O}}$ is baseline and $\mathbf{p}_{x,\mathrm{V}} - \mathbf{p}_{x,\mathrm{O}}$ is difference of principal point offset.

Now the virtual view $\mathbf{I}_{\mathrm{V}}$ is generated using either (3.12) or (3.13) from the reference view $\mathbf{I}_{\mathrm{L}}$ and the corresponding depth information $\mathbf{Z}_{\mathrm{L}}$. Usually, projected pixels do not yield integer values. As a result these positions do not belong to the virtual view grid. This problem is also known as the re-sampling problem. Therefore projected pixels are rounded to the nearest integer and then the color and depth values of reference pixels are transferred into the virtual view pixels. Moreover, occlusions in the virtual view cause projection of more than one pixel into the same location in the virtual view. In this case, the pixel closer to the camera is selected.

3.3.2 Backward warping

Unlike forward warping, backward warping produces a virtual view by obtaining the virtual view depth map. In this approach, the depth maps are first warped to the virtual view position using forward warping. Then the warped depth map is processed by filling the missing information. Finally, the warped depth map is used to locate corresponding original view pixels for each virtual view pixel and then the texture of original view is mapped to virtual view. Re-sampling issues with forward warping can be reduced by using the backward warping approach, since each virtual view pixel is located in the original view.

3.3.3 Merging

The last two sub Sections 3.3.1 and 3.3.2 presented virtual view generation from a single V+D. This section presents the view interpolation, which is how the intermediate view is synthesized using MVD. Firstly, two reference views are warped to a desired view position. Next, the two warped views are merged into one view. As more information is used from two views the numbers of artifacts are minimized, since they are partially filled with other warped views. Literature presents several ways to merge the views [TLLG09]. The common way to blend the views is by giving priority to the nearest view and the blending process becomes:

$$\mathbf{I}_{\mathrm{V}}(\mathbf{m}) = w\mathbf{I}_{\mathrm{L}}(\mathbf{m}) + (1-w)\mathbf{I}_{\mathrm{R}}(\mathbf{m}), \tag{3.14}$$

$$w = \frac{|\mathbf{t}_{\mathrm{V}} - \mathbf{t}_{\mathrm{R}}|}{|\mathbf{t}_{\mathrm{V}} - \mathbf{t}_{\mathrm{L}}| + |\mathbf{t}_{\mathrm{V}} - \mathbf{t}_{\mathrm{R}}|}, \tag{3.15}$$

where $\mathbf{t}$ is the translational vector of a camera and the subscripts L, R and V represent the left, right and virtual cameras respectively and w is the linear weight, which depends on the distance between the reference and the virtual view translation vector.

3.4 Rendering artifacts

The DIBR method has limitations due to inherent artifacts in the warped image. As a result, perception of the depth and the desired experience are affected due to the reduced visual quality of virtual view. Rendering artifacts are severe in the extrapolated view, since there is no additional information to handle, whereas in the intermediate view the artifacts are partially filled with the information from the other warped view. An example of an extrapolated view using DIBR is shown in Fig. 3.7. Possible artifacts in the warped images are classified as shown in Fig. 3.6. The artifacts in the warped image are ghosting and holes (a.k.a. uncovered areas).

Ghosting artifacts are mixtures of colors at the edges in the original image which are projected into the neighboring objects in the warped image. The cause of the

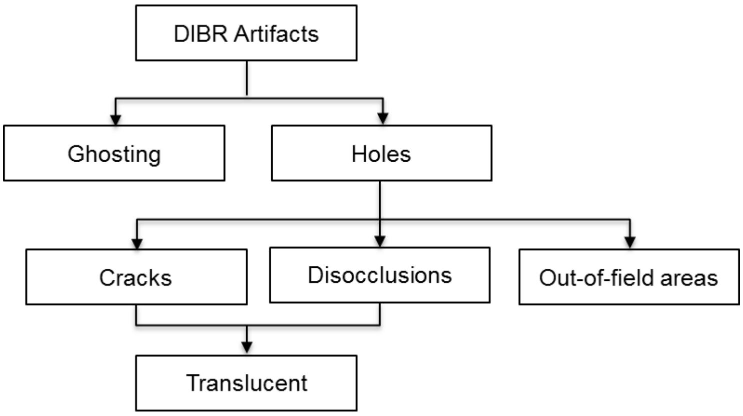


Figure 3.6: DIBR artifacts classification.

ghosting artifact is the depth and texture misalignment and mixing of neighboring colours at the depth discontinuities (see Fig. 3.9(e)).

Holes are undefined pixels in the rendered images. They appear due to uncovered (not captured) regions because they were occluded by foreground in the original view. They are classified into three types, namely cracks, disocclusions and out-of-field areas.

Cracks are one to two pixel-wide empty regions in the warped image. Cracks appear in the virtual view due to assigning warped pixel coordinate positions to the nearest integer coordinates (see Fig. 3.9(a)).

Translucent cracks appear for the same reason as the *cracks*, but these areas possess background information as consequence of occlusions in virtual view (see Fig. 3.9(d)).

Disocclusions occur at depth discontinuities near the object borders. Disocclusions are the result of regions being revealed in the virtual view as they have been occluded by the foreground in the original view. Therefore occlusions in the original view translate into disocclusions and appear as holes in the virtual view as the result of the warping process. When there are more than two depth layers, we define foreground and background as relative terms. Occlusion areas between a relative foreground-background pair are referred to as overlaid occlusions. An example of relative foreground-background case is shown in Fig. 3.8(c), where the woman's body is a relative background to her hand, whereas the wall is an absolute background. These definitions are later considered in Chapter 8. It is worth noting that the size of the disocclusions depends on the size of the baseline and the scene depth (see Fig. 3.9(b)).

Translucent-disocclusions Translucent disocclusions differ from common disocclusions by exposing texture information that is present behind the relative background. Translucent disocclusions occur only when the depth has three or more layers, and where there are occlusions between layers. An example of this case is shown in Fig. 3.8(a) in the area near the woman's hand. Note in Fig. 3.8(c) that the wall tex-

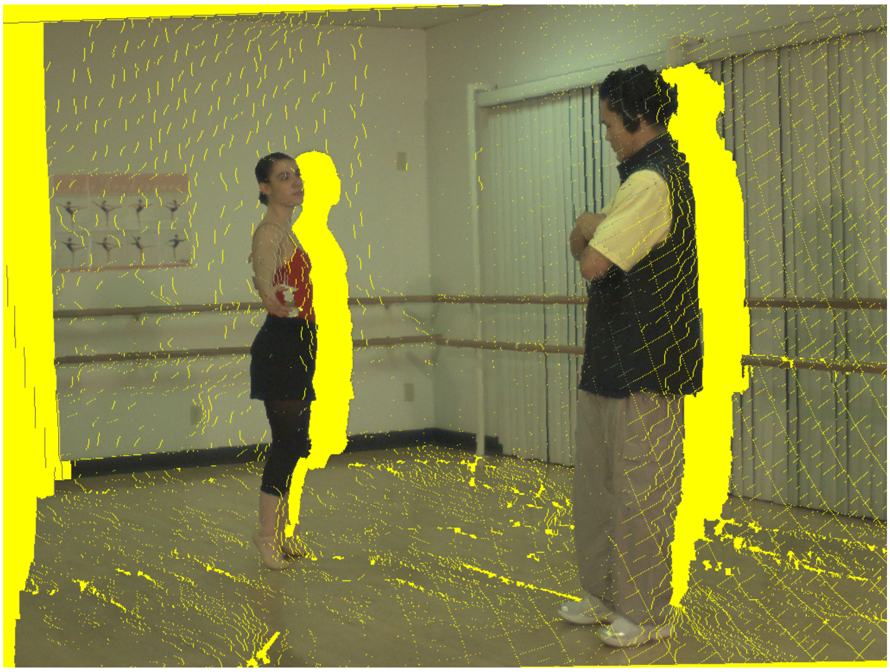


Figure 3.7: Warped Image using DIBR.

ture is seeping through the woman's body, where the hand and body are relative foreground and background respectively. The magnitude of the disturbance created by translucent disocclusion artifacts depends on the placement of occlusions in the scene and the occlusion area. However, translucent disocclusion artifacts need to be addressed since they can deteriorate the perceived visual quality.

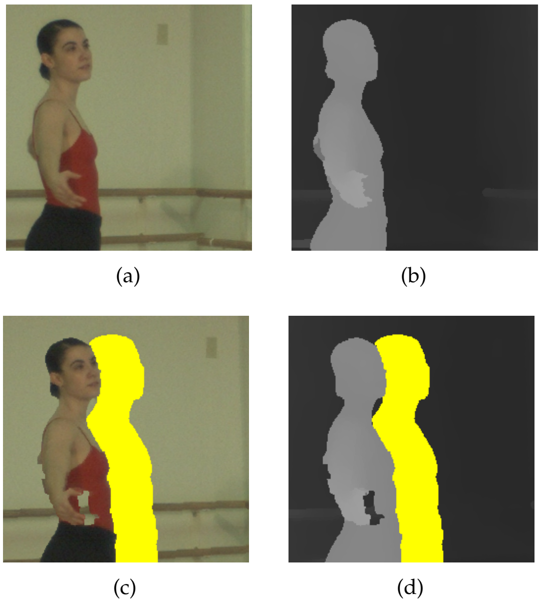
Out-of-field areas are also holes, which occur at image borders. They occur due to the limited field of view in the original view (see Fig. 3.9(c)).

Unnatural contours are the pixilation of "new" edges between background and foreground in the warped image due to the colors of the pixels at the edge not being blended.

To reduce the DIBR artifacts a number of state-of-the-art solutions have been introduced. The reference works are presented in Chapter 5. However, disocclusion handling still remains a challenging problem.

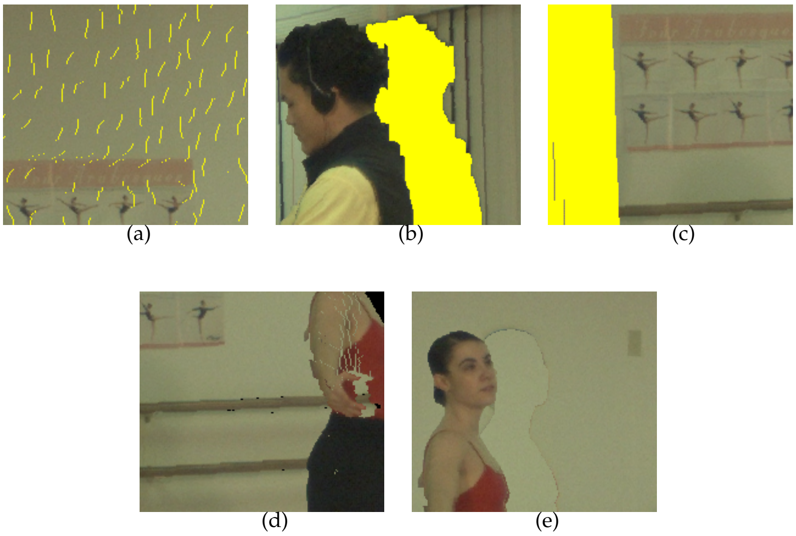
3.4.1 Contributions

The contribution of the author to this chapter is defining the translucent disocclusions in rendering artifacts. The details of how these artifacts are identified and handled are presented in Chapter 8.



(a) (b) (c) (d)

Figure 3.8: Illustration of translucent disocclusion: (a) Texture image; (b) Depth image; (c) Warped image; (d) Warped depth image (disocclusions in yellow).



(a) (b) (c) (d) (e)

Figure 3.9: Rendering artifacts: (a) Cracks; (b) Disocclusion; (c) Out-of-field area; (d) Translucent cracks are above the woman's palm and translucent disocclusions are visible near woman's palm,where the background is appearing through her body; (e) Ghosting areas are thin lines around the disocclusions.

Chapter 4

Image Inpainting

Image inpainting is a method to fill missing or damaged regions in an image using information from the surrounding areas. Image inpainting has been an active topic in image processing over the years, with a wide spread of applications. These include image restoration of damaged photographs, transmission error recovery in the context of compression, object removal in the context of image editing, image up scaling in the context of content generation, and disocclusion filling in the context of rendering virtual views from a single or several views.

The purpose of this dissertation is to investigate rendering problems and to facilitate high quality content generation. Holes are one of the major problems of DIBR in the context of 3D content generation. An understanding of the basic principles of image inpainting is necessary to fully grasp the contributions of this dissertation. This chapter provides background knowledge for different types of image inpainting techniques. The information provided in this chapter summarizes the main parts of the inpainting process, relevant to the contributions of this thesis. Interested readers can look into the relevant literature for each topic for more detailed descriptions.

4.1 Overview of image inpainting

Originally, inpainting was referring to manually painting defects such as cracks or missing areas in old photographs. But in the digital context inpainting is the process of automatically recovering missing information from the available information in the surrounding areas [BSCB00]. In the context of this dissertation, the missing information is referred to as a *hole*. Image inpainting is also known as image-completion, texture synthesis and hole-filling. Image inpainting methods can be broadly classified into three types: textural, structural and hybrid inpainting methods (see Fig. 4.1). A good review on image inpainting methods can be found in [GM14].

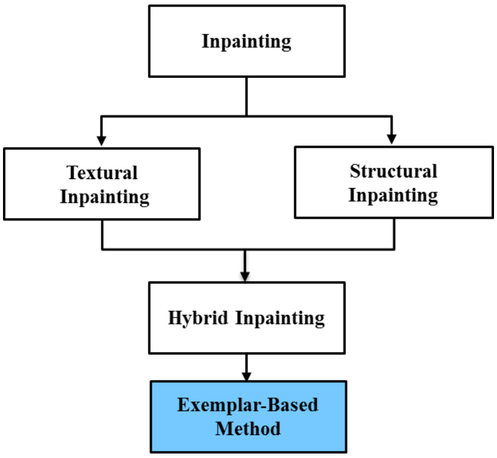


Figure 4.1: Image inpainting methods classification.

Textural inpainting methods replicate the repetitive patterns in the image, which surround the holes. Many of these methods use Markov Random Field (MRF) to model the local patterns using small amount of known textures around holes. They synthesize textures by sampling and copying textures from neighboring regions to fill the holes [EL99, Har01]. Texture inpainting methods perform well for homogeneous texture images. However, they are poor in handling real world images, since real world images consist of a mixture of different textures and linear structures. Structural inpainting methods identify the structure details in the neighborhood and propagate the information using a diffusion process, hence they are called diffusion-based methods. Hybrid inpainting methods combine the advantages of both textural and structural methods.

The overall idea of structural inpainting methods is to identify the intrinsic geometry information of an image and then perform a smooth continuation of the structures by using partial differential equations (PDEs) or variational methods. The local geometry is extracted by using the isophote or structure tensors. Isophotes are defined as level lines, which have constant intensity (smallest spatial variation), meaning that they are normal to the image gradient, as the gradient direction gives the direction of the largest intensity change. Isophotes are denoted as $\nabla\mathbf{I}^{\perp}$, where $\nabla\mathbf{I}$ is the gradient of an image $\mathbf{I}$. The isophote direction is shown in Fig. 4.3(a). Another way to extract the geometry is by using spectral elements of the structure tensor (a.k.a. Dizenzo matrix), which is given by $G = \nabla\mathbf{I}\nabla\mathbf{I}^T$ [Zen86]. The Eigen values and their corresponding vectors denote the strength and orientation of the variations in an image. The largest variation direction corresponds to the gradient, which is analogous to the edge normal. The lowest variation direction corresponds to the isophote direction, which is analogous to the edge tangent.

Hybrid methods are also known as exemplar-based methods, which copy exemplars (data patches) to fill holes. The inpainting methods used in the context of DIBR are diffusion-based methods and exemplar-based methods. These methods are presented in Sections 4.2 and 4.3 to the extent required for this thesis. For a more in-depth review on the applications of hole-filling methods in the context of DIBR see [TLD07].

4.2 Diffusion-based methods

In this section, an overview of diffusion methods is presented taking an evolutionary approach. First, the basic idea of diffusion and image regularization are presented, since the principles of diffusion were initially used in the context of image regularization. Then diffusion-based inpainting methods are presented. Finally, variational inpainting methods are briefly described.

Diffusion is defined as propagation of local geometry information with smoothing constraints. The term "diffusion" comes from physical processes such as the heat propagation in solid structures. Diffusion is expressed using partial differential equations (PDE). Image regularization (a.k.a. smoothing) is defined as finding smooth surfaces, similar enough to the original noisy image, by minimizing variations in the image using either PDE or energy formulations [Tsc02]. The heat diffusion equation in general form is given as:

$$\begin{aligned}\frac{\partial\mathbf{I}}{\partial t} &= \nabla\cdot\left(D_{\text{diff}}\cdot\nabla\mathbf{I}\right),\\ \mathbf{I}_{(t=0)} &= \mathbf{I}_0,\end{aligned} \tag{4.1}$$

where, $\mathbf{I}_0$ is a degraded image in the image regularization context, and image with holes (see Fig. 4.3) in the inpainting context. The PDE evolution $\frac{\partial\mathbf{I}}{\partial t}$ indicates the continuous improvement of $\mathbf{I}$ with time. $\nabla\cdot$ represents the divergence operation, which measures the image intensity variations in x and y directions. D_{diff} is diffusion coefficient [1]. The type of diffusion depends on D_{diff}. If D_{diff} is a positive scalar value then the diffusion is called an isotropic diffusion, performed in all directions by minimization variations. If the diffusion coefficient is not changing over the image,

[1] The symbol D is used for data term in the exemplar inpainting method to be consistent with the inpainting notation, therefore we used D_{diff} for the diffusion coefficient instead of the conventional notation D

it is called a linear diffusion. If D_{diff} is a tensor, a positive symmetric matrix, the diffusion is called anisotropic diffusion and is performed depending on different weights given to spatial directions [Wei98].

The linear PDE-diffusion minimizes variations equally in all directions, which causes blurring at the edges. To avoid the diffusion across edges, a non-linear diffusion was introduced by using diffusion coefficient as a function of the gradient which controls the diffusion. Although this method is a non-linear diffusion, sometimes it is also referred to as anisotropic diffusion [Wei98]. Later, anisotropic diffusion methods were introduced by introducing the diffusion tensor. The diffusion tensor is expressed by the spectral elements of the structure tensor in order to weight the diffusion in spatial directions so that the diffusion flows along the isophotes [Wei96]. Subsequently, vector valued anisotropic diffusion was presented [Tsc06].

The concept of digital inpainting using diffusion was first introduced by Bertalmio et al. [BSCB00]. The method was based on the idea of propagating both geometric and photometric information that hit the border of the holes into the missing regions. This method used an anisotropic model to propagate image Laplacian (smoothness priors) into the missing regions along the isophote direction iteratively by using

$$\frac{\partial \mathbf{I}}{\partial t} = \nabla \left(\Delta \mathbf{I} \right) \cdot \nabla \mathbf{I}^{\perp}, \tag{4.2}$$

where $\Delta \mathbf{I}$ is the image Laplacian, which estimates the amount of smoothness. At the steady state, the update term becomes zero which implies no variations in the isophote directions. Subsequently, another diffusion method was presented with an analogy to the fluid dynamics. This diffusion method aims at preserving the isophote direction [BBS01]. Further a fast processing diffusion method was introduced [Tel04].

A different type of PDE-based inpainting is based on the variational minimization of energy functions. Chan et al. presented a total variational inpainting model for non-textural images [SC02]. This is a non-linear diffusion method, which is based on the strength of the isophote. Although, the total variational inpainting method is effective for recovering sharp edges, it has problems in connecting the broken edges. The total variational model was later extended to the Curvature Driven Diffusion model (CDD), which includes the geometry of the isophote along with the strength to handle connection problems [CS01].

$$\frac{\partial \mathbf{I_p}}{\partial t} = \nabla \cdot \left(\frac{g(|k_{\mathbf{p}}|)}{|\nabla \mathbf{I_p}|} \nabla \mathbf{I_p} \right), \tag{4.3}$$

where $k_{\mathbf{p}} = \nabla \cdot \left(\frac{\nabla \mathbf{I_p}}{|\nabla \mathbf{I_p}|} \right)$ is the curvature of the isophote through a pixel $\mathbf{p}$, and g is the control function for diffusion in order to connect the broken edges. However, these methods have problems in handling textural images [SC02].

The above mentioned image regularization techniques using diffusion tensors can also be applied to hole-filling in the inpainting context. All regularization methods iteratively fill holes using respective diffusion or variational models. However, these methods mainly focus on the structure propagation and hence cannot perform

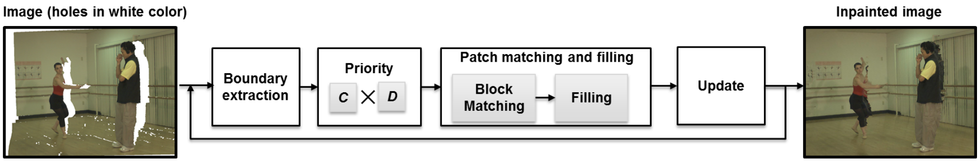


Figure 4.2: Exemplar-based inpainting method.

well for texture filling and consequently suffer from blurring problems. Moreover, inpainting is an ill-posed problem since no details about the hole pixel values is available. Exemplar-based methods are among the feasible solutions, since they aim at reconstructing both the structure and texture of the missing regions.

4.3 Exemplar-based methods

To replicate structure and textural details, exemplar methods combine the textural and structural inpainting methods and so are termed hybrid inpainting methods. An initial attempt at exemplar inpainting was introduced by Harrison et al. [Har01]. This method computes the hole filling order by using the amount of the texture detail in the neighboring regions and then performs texture transfer from the surrounding regions for the filling step. However, the quality of the inpainted image depends on the order in which the filling is performed. Criminisi et al. presented an efficient filling order based on the isophote linear structures in the neighboring region [CPT04]. The general idea of the Criminisi et al. method is illustrated in Fig. 4.2 and the processing steps in this method are as follows: (i) A filling order is computed on the hole boundary using a priority term, which consists of structure details and the amount of known pixel information in the neighborhood. (ii) A patch (a.k.a. target patch) is selected at the highest priority value on the boundary, and then similar data is searched in the neighborhood using known values in the patch. The best data patch is referred to as the source patch. (iii) Finally, holes in the target patch are filled with the source patch and the confidence term is then updated. The whole process is repeated until the whole hole is filled.

Assume an image $\mathbf{I}$ having a hole region Ω that needs to be inpainted. The remaining portion of the image, apart from the hole region, is defined as the source region $\Phi = \mathbf{I} - \Omega$, which is used to find a source patch for filling holes in the target patch. The boundary of the hole is defined as $\delta\Omega$ (see Fig. 4.3). The boundary of the hole can be divided into foreground and background. A set pixels on the boundary that belongs to the background is referred to as background boundary, and pixels on the boundary that belong to the foreground are referred as foreground boundary. The image that represents the hole is referred to as hole mask.

The hole boundary is extracted by applying Laplacian on the hole mask (where the position of holes are represented as ones), and then priorities are computed on the boundary by taking a patch data centered at each pixel on the boundary. Let

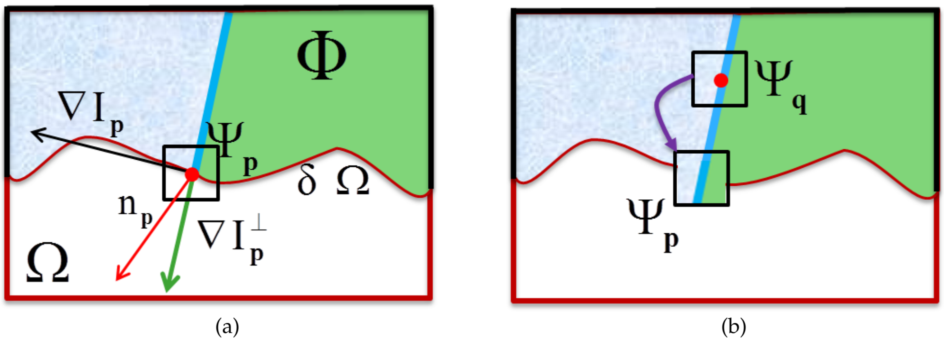


Figure 4.3: Exemplar based inpainting notation (white color represents hole regions): (a) Image with holes and inpainting notation; (b) Illustration of patch filling.

a patch $\Psi_{\mathbf{p}}$ be centered at a pixel $\mathbf{p}$. The priority term is defined as the product of confidence and data terms $P(\mathbf{p}) = C(\mathbf{p}) \cdot D(\mathbf{p})$, where $C(\mathbf{p})$ is the confidence term and $D(\mathbf{p})$ is the data term. The confidence term computes the percentage of non-hole pixels in a patch, and the data term extracts the local geometry, isophotes in the neighborhood, defined as:

$$C(\mathbf{p}) = \frac{1}{|\Psi_{\mathbf{p}}|} \sum_{\mathbf{q} \in \Psi_{\mathbf{p}} \cap \Omega} C(\mathbf{q}), \tag{4.4}$$

$$D(\mathbf{p}) = \frac{\langle \nabla^{\perp} \cdot \mathbf{n}_{\mathbf{p}} \rangle}{\alpha}, \tag{4.5}$$

where $|\Psi_{\mathbf{p}}|$ is the area of $\Psi_{\mathbf{p}}$ (in terms of number of pixels) and α is a normalization factor. $\mathbf{n}_{\mathbf{p}}$ is a unit vector orthogonal to $\delta\Omega$ at a point $\mathbf{p}$, and $\nabla^{\perp} = (-\partial_y, \partial_x)$ is the direction of the isophote, which is orthogonal to the gradient. Initial value of the confidence term for hole pixels in Ω is $C(\mathbf{q}) = 0$ and for source region pixel in Φ is $C(\mathbf{q}) = 1$.

Once the target patch $\Psi_{\hat{\mathbf{p}}}$ centered at $\hat{\mathbf{p}}$ is selected to be filled, then block matching is performed using known pixels in the target patch within the source region:

$$\Psi_{\mathbf{q}} = \arg \min_{\Psi_{\mathbf{q}} \in \Phi} \left\{ \|\Psi_{\hat{\mathbf{p}}} - \Psi_{\mathbf{q}}\|_2^2 \right\}, \tag{4.6}$$

where $\|\Psi_{\hat{\mathbf{p}}} - \Psi_{\mathbf{q}}\|_2^2$ is the sum of squared difference between known pixels in two patches $\Psi_{\hat{\mathbf{p}}}$, $\Psi_{\mathbf{q}}$. Once the source patch $\Psi_{\mathbf{q}}$ is found, the hole pixels in the target patch are filled from the source patch $\Psi_{\mathbf{q}}$. Then the confidence term is updated as:

$$C(\mathbf{q}) = C(\hat{\mathbf{p}}), \forall \mathbf{q} \in \Psi_{\hat{\mathbf{p}}} \cap \Omega. \tag{4.7}$$

The hole-filling process continues until all the hole pixels in the image are filled.

A set of images inpainted with the diffusion-based method and the exemplar-based method are shown in Fig. 4.4. In the image inpainted by the diffusion-based

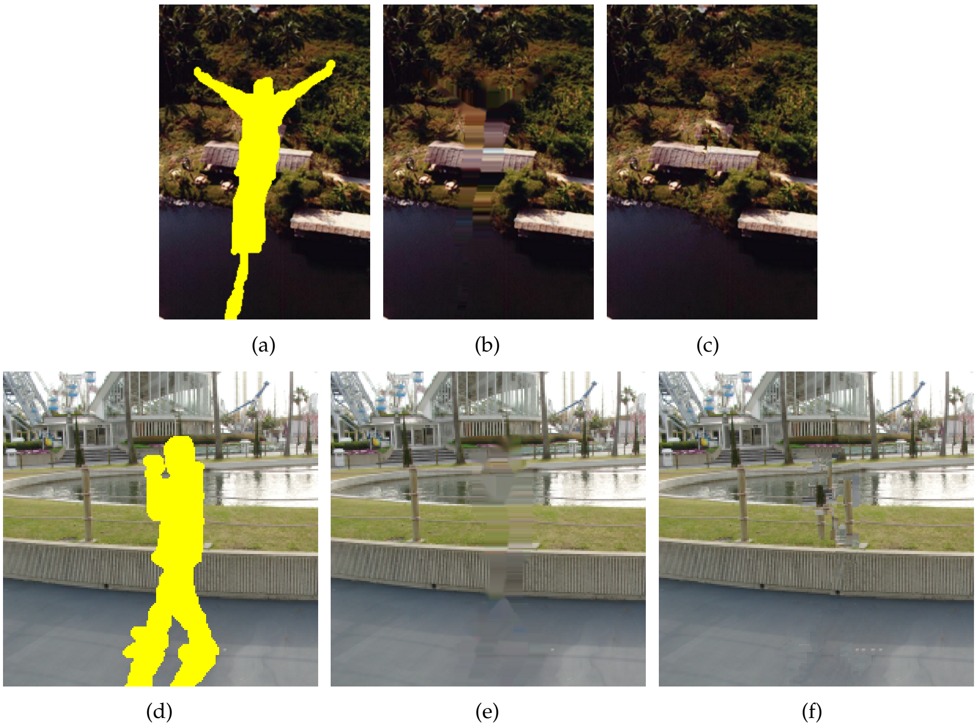


Figure 4.4: Diffusion and exemplar-based inpainting methods results: (a), (d) Test images (holes are in yellow); (b), (e) Hole-filling with diffusion-based inpainting [BBS01]; (c), (f) Hole-filling with exemplar-based inpainting [CPT04].

method, the structure is recovered partially but the holes still suffer from blurring. Whereas in the image inpainted by the exemplar-based method, the textures are produced along with the source textures (see Fig. 4.4(b) and (c)). Several variations of exemplar-based methods are proposed in literature, improving the filling order by using various priority terms and different patch matching strategies. This thesis is mainly focused on hole-filling in the context of virtual view synthesis. For more details and variations on the exemplar-based methods see the works in [AFCS11, GM14].

Chapter 5

Related Work on View Rendering

Chapter 3 and 4 presented an overview and basics of DIBR and image inpainting. This chapter gives an overview of the related work in the context of view rendering methods. The literature survey consists of two parts. The first part presents methods that address DIBR artifacts in general. The second part presents hole-filling methods that address the holes in the rendered view.

5.1 Overview

Several methods have been introduced in the view rendering literature in order to address the rendering artifacts. These methods are classified into two groups: general rendering methods, which address DIBR artifacts in the common way and hole-filling methods, which address the holes in the rendered view using various filling techniques. These methods are presented in the following Sections 5.2 and 5.3.

5.2 General rendering methods

General rendering methods can be divided into pre-processing and post-processing methods; the names indicate how the problems are handled. Pre-processing methods apply various techniques before warping and post-processing methods apply different tools after warping. There are also a number of methods that combine both pre-processing and post-processing techniques to address rendering problems. These methods are also known as hybrid methods.

5.2.1 Pre-processing

One of the most common ways to reduce minor rendering artifacts is pre-processing the input depth map (see Fig. 5.1(a)). Chapter 3 explains that artifacts in the rendered view are due to errors or depth discontinuities in the depth map. Thus pre-processing methods apply smoothing techniques on the depth map to reduce rendering artifacts. Gaussian or median filters are commonly used to smooth the depth maps [Feh04, TAZ$^+$04]. However, filtering the depth map regardless of important information such as edges, may lead to structure degradation.

To counter the problems with symmetrical smoothing filters, Zhang et al. introduced asymmetrical filter smoothing, which controls smoothing in the horizontal direction, so the structures are preserved [ZT05]. Filtering the entire depth map increases blurring. Therefore, edge dependent filtering methods are introduced in [ZwPSxZy07, DTPP07]. Later, adaptive filters were introduced, which steer the smoothing based on gradient direction, unlike fixed filters, and preserve discontinuity [PJO$^+$09].

5.2.2 Post-processing

Pre-processing methods reduce artifacts, however, hole-filling techniques are still required to fill the holes in the warped images. Moreover, geometrical distortions are one of the consequences of pre-processing the depth map. Thus a post-processing step becomes necessary after warping. Commonly, cracks are handled by simple interpolations and backward warping. Ghosting artifacts are usually handled by removing the few pixels around the background region at depth discontinuities. As mentioned in Chapter 3, backward warping is one solution to reduce small artifacts in the pre-processing stage, followed by hole-filling (see Fig. 5.1(b)). [MFY$^+$09, TLLG09].

5.2.3 Hybrid-processing

Layering method

Zitnick et al. introduced a layering method to reduce artifacts [ZKU$^+$04]. In this method, the input data is divided into different layers based on the reliability criteria. Subsequently, layers of each view are warped to the virtual view position and merged using depth ordering and weighting. A post processing step is performed at the end to reduce the remaining artifacts. The block diagram of the layering method is shown in Fig. 5.1(c). Because of artifacts which still exist in the rendered view, the method was improved by a layering approach with different layering constraints [MSD$^+$08]. Later, this method was further improved in the step "selection of layers" [SHKO11].

View synthesis reference software

MPEG has developed software that renders virtual views using V+D and MVD data. The software is known as View Synthesis Reference Software (VSRS). The software consists of several tools to reduce artifacts in the virtual views. VSRS requires texture, depth and camera parameters in order to generate the virtual view position. It has two different modes to render views: general and 1D mode. Backward warping and diffusion based inpainting are used in the general mode, whereas forward warping and background propagation are used in the 1D mode. Input views are also up sampled to get high precision values. Moreover, several merging strategies are employed to combine the data from different views in view interpolation to produce intermediate views. This consists of both pre-processing and post processing steps. The block diagram of VSRS 1D mode is shown in Fig. 5.1(d). A detailed description of VSRS is presented in [vsr10]. Improvements to VSRS in 1D mode using reliability map creation is presented in [Hhi12]. Although VSRS shows several improvements, rendered views still exhibits artifacts.

5.3 Hole-filling

A number of solutions have been introduced in the literature to fill the holes in warped images. Hole-filling methods can be categorized into *interpolation*, *diffusion-based inpainting* and *exemplar-based inpainting*.

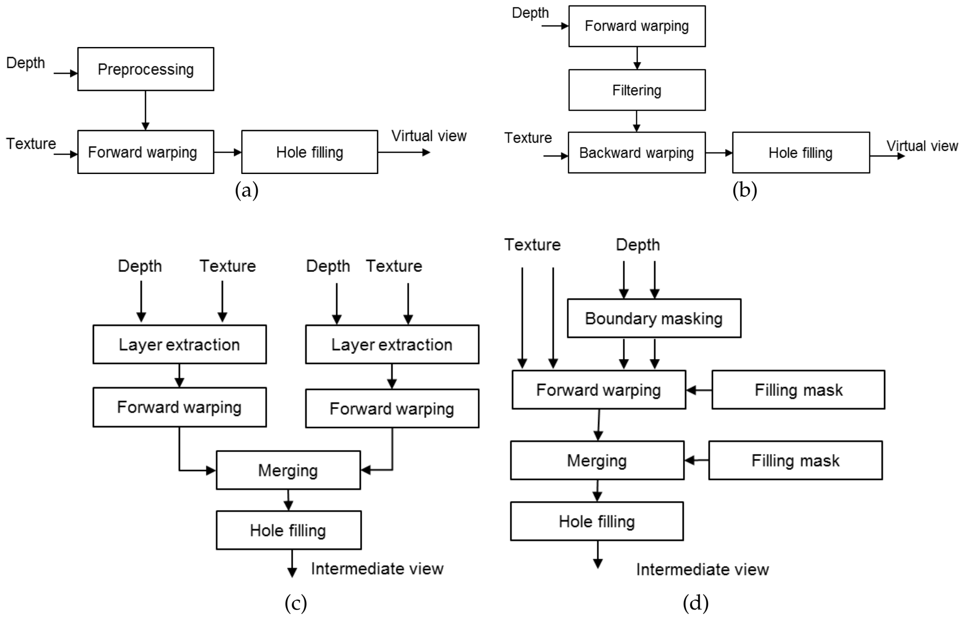


Figure 5.1: Existing rendering approaches: (a) Pre-processing; (b) Backward warping; (c) Layering method; (d) VSRS.

5.3.1 Interpolation

Interpolation methods use the texture from pixels on the border of the hole in order to reconstruct the textures for the pixels inside the hole. As these methods only rely on border pixels, the visual quality of a rendered image depends on the virtual camera position and textures in the image. For a virtual camera farther away from the original camera, i.e. large baseline B, and for large depth discontinuities, the holes in the warped images are bigger, and applying interpolation hole-filling method results in unnatural and inconsistently filled areas. Moreover, these methods cannot reconstruct any structural details, so that artifacts are easily noticeable.

5.3.2 Diffusion-based inpainting

A brief overview of the inpainting process is presented in Chapter 4. Similar to interpolation methods, diffusion-based methods also rely on boundary pixels of the hole to reconstruct the missing regions in the hole area [BSCB00, Tel04]. However, the filling process is guided by the structure details in the neighborhood, so that the structure can be replicated in the missing regions. However, diffusion methods also share disadvantages such as blurring and lack of textures in the filled regions. Moreover, blurring increases with the size of the hole to be filled. The VSRS also uses diffusion-based inpainting to recover the hole regions [vsr10]. Diffusion-based inpainting in the context of view rendering suffers from another problem as well, which is when foreground is propagated into the holes, which appears when the foreground side of the hole is allowed to propagate. To avoid foreground propagation problems, Oh et al. [OYH09] filled the boundary of the holes along the foreground side using the background information on the opposite side of the hole, and then applied diffusion. However, the inherent problems with diffusion methods still exist and are visible depending on the size of the hole and the texture in the view.

5.3.3 Exemplar-based inpainting

Exemplar-based methods can efficiently reconstruct both the structure and texture details of the holes [CPT04]. Therefore, several exemplar-based methods have been developed with different constraints. To avoid the foreground propagation into holes it is very important to consider the depth in the filling process. To follow and understand the evolution in the exemplar-based inpainting methods, related work in the context of view rendering are further divided into three categories, if the utilized depth map for the inpainting process is a true (depth at the virtual camera), filled-depth and warped depth map. In the case of true depth it is assumed that depth values of the hole pixels are given. In filled-depth, the holes in the depth image are filled before the texture image. Finally, in the case of the warped depth, both texture and depth map are filled simultaneously.

True depth

Daribo et al. use depth information in the inpainting process [DS11]. Moreover, they introduce a depth regularity term in the priority term: $P\left(\mathbf{p}\right) = C\left(\mathbf{p}\right) \times D\left(\mathbf{p}\right) \times \hat{L}\left(\mathbf{p}\right)$ where the depth regularity term at $\mathbf{p}$is given by:

$$\hat{L}\left(\mathbf{p}\right) = \frac{|\mathbf{Z_p}|}{|\mathbf{Z_p}| + \sum_{\mathbf{q} \in \mathbf{Z_p} \cap \Phi} \left(\mathbf{Z_p}(\mathbf{q}) - \overline{\mathbf{Z_p}}\right)^2}. \tag{5.1}$$

This term is approximation of the inverse variance of depth values in the depth patch $\mathbf{Z_p}$ centered at $\mathbf{p}$. This means that the term $\hat{L}$ gives the highest priority to a target patch containing information from the background over a patch containing foreground information. Depth is also added in the block matching process while searching for the best source patch using a weighting process. This method improved the visual quality compared to the exemplar- based method [CPT04], which does not consider depth but fills the holes with foreground information because of the filling order.

Gautier et al. also used the true depth information in the inpainting process [GMG11]. In their work, the structure tensor J data term is computed instead of simple gradients:

$$J = \sum_{l=R,G,B,Z} \nabla \mathbf{I}_l \nabla \mathbf{I}_l^T, \tag{5.2}$$

$$D(\mathbf{p}) = a + (1 - a) \exp \left(\frac{-C_1}{(\lambda_1 - \lambda_2)^2} \right), \tag{5.3}$$

where $\nabla \mathbf{I}_l$ is the local spatial gradient over a 3x3 window. J is the 3D structure tensor and λ_1 and λ_2 are the Eigen values of J and C_1 is a positive constant value and $a \in [0, 1]$. Using the Eigen values information, which determines the largest and smallest variations, the inpainting process is performed along the direction of the lowest variation. Moreover, priority is only computed on the background hole boundary by using the position of the warping views. Then average values of few best patches are used to fill the holes instead of only one patch, based on the idea in [WSI07].

Filled-depth

To improve the inpainting quality, Jantet et al. filled the warped depth map using the background information and then used the pre-processed depth map in the inpainting process [JGM11]. Further, Lim et al., introduced foreground-background classification of depth into the inpainting methods to inpaint holes with background information [LKL$^+$11].

An alternative formulation of the layered inpainting method is proposed by Wolinski et al. in order to provide inter-view consistency [WMG13]. The basic steps of this method are as follows: First the depth is classified into several foreground-background layers and the disocclusions between the layers are located. Next, holes in the depth map are filled. Using the depth map, layers are inpainted. Finally, inpainted layers are projected into the virtual views. The method creates consistency between views, but holes are not spatially consistent due to the selection of classification and the inpainting method. Moreover, inpainted images still exhibit translucent artifacts and foreground propagation into holes.

Warped depth

Ahn et al. use Hessian-based data terms to find the structure details during the inpainting process, similar to [GMG11]. Moreover, depth classification is performed locally during the inpainting in order to propagate only background information [AK13]. Despite these improvements, the inpainted views still exhibit spatial texture inconsistencies due to the inconsistent foreground-background classification and computing filling order, which is limited to only the horizontal direction.

Despite the improvements and new advances, the rendered views still suffer from spatial inconsistencies. Consequently, the visual quality is reduced.

Chapter 6

Edge-Aided Depth-Image-Based Rendering

The previous chapter presented related works regarding depth-image-based rendering (DIBR). As the methods require specific processing steps for handling artifacts, this chapter presents an alternative rendering method, which is edge-aided depth-image-based rendering for 3D video formats V+D and MVD with both subjective and objective evaluations. Furthermore, this chapter presents a study of DIBR artifacts and their remedies by comparing the proposed method with previous methods.

6.1 Introduction

DIBR methods produce a number of artifacts including holes (cracks, disocclusions, out-of-field areas), ghosting and translucent artifacts, which are detailed in Section 3.4. These artifacts reduce the visual quality of the virtual view (see Fig. 6.1). Several state-of-the-art methods which provide dedicated post-processing solutions to correct each artifact are presented in Chapter 5. So the question arises if there is a more fundamental approach that avoids DIBR artifacts in the virtual views rendered from 3DV formats: V+D and MVD.

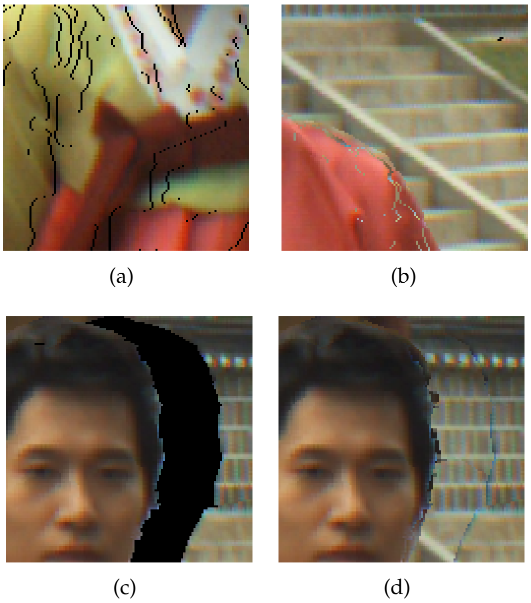

(a)
(b)
(c)
(d)

Figure 6.1: Rendering artifacts: (a) Cracks; (b) Translucent cracks; (c) Disocclusion marked in black color; (d) Ghosting .

An edge-aided rendering method is introduced as a straightforward approach for avoiding the rendering artifacts. The proposed method has two main novelties: (i) Introducing edge-pixels to keep track of both the foreground and background information at the edges, whereby interpolation is straightforward. (ii) Merging two rendered views using the projected pixels' actual positions such that holes are first filled by any data available in the two adjacent views, and any remaining hole pixels are interpolated from background color information.

This chapter is organized as follows: Sections 6.2 and 6.3 explains the proposed edge-aided rendering method for view extrapolation and view interpolation. The methodology for evaluation described in Section 6.4. The results and analysis are presented in Section 6.5. Finally, Section 6.6 concludes this chapter.

6.2 Proposed method for view extrapolation

The proposed method relies on the fundamental principles of 3D warping as follows: The pixels that are projected into the virtual view remain in the exact same position in the first stage, and are not assigned to the closest virtual pixel as is commonly done in DIBR methods. A subsequent interpolation of projected samples gives the values of the pixel grid in the virtual view. Specific edge-pixels are introduced at the borders that lie between the foreground and background. They are assigned foreground depth but contain both background and foreground colors at the corresponding side of edge. These edge-pixels keep track of the placement of edges and the color of the respective side, and so avoid smearing out edges over disocclusions in the interpolation process. Further edge softening is applied by a low-pass filtering of edges.

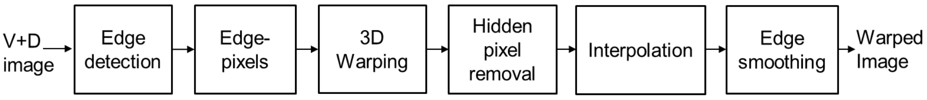


Figure 6.2: Edge-aided rendering for view extrapolation from V+D.

Fig. 6.2 shows a block diagram of the edge-aided rendering method for the V+D format. The assumption of only horizontal displacement reduces the warping equation to a one-dimensional translation given by (3.13). The proposed method thereby turns into a line-by-line algorithm with the following steps depicted in Fig. 6.4.

Edge detection: The difference between neighboring pixels on the depth map is calculated, which identifies the edges between the foreground and the background or objects at intermediate levels. Because the detection is made along one line, the detection is narrowed down to finding a difference that is larger than a threshold.

Edge-pixels: The so called *edge-pixel* has a position and color value that is similar to a general pixel, but the position is a rational number, i.e. the position of the edge is horizontally shifted by a small amount. An illustration of an edge-pixel is shown in Fig. 6.3. The edge-pixel, which is introduced at the depth discontinuities, depends on the warping direction. For example, a right warped view has disocclusions on the right side of the foreground objects, thus the edge-pixel $(7 + \epsilon)$ is introduced on the right side of the foreground (see step1 in Fig. 6.4 at pixel position 7). The edge-pixel contains foreground depth and background color, so that it follows the foreground (see step2 in Fig. 6.4 edge-pixel position after warping at the pixel position 5.11). In general, images consist of mixed colors of both foreground and background at object boundaries over transition regions. The transition range is about one or two pixels wide due to the averaging of colors in the camera pixel sensor. Hence, background color values for the edge pixel are selected outside the transition area in order to avoid smearing colors over a larger area.

3D Warping: Applies forward warping to find the new floating point coordinates for each pixel in the virtual view.

Handling Hidden pixels: Once the pixels from the original view are warped to a

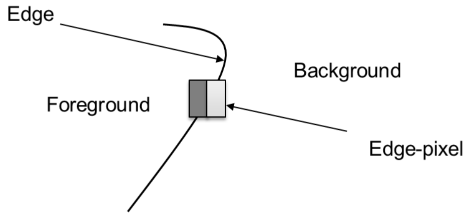


Figure 6.3: Edge-pixels are introduced at borders between foreground and background. They contain only foreground depth but both background and foreground colors at the corresponding side of the pixel.

new view point, there are a few pixels that become occluded and appear through the cracks in the virtual views. They are identified after warping by taking the depth differences between the neighboring pixels. After identifying the *hidden pixels* (see step2 in Fig. 6.4 at pixel position 3.6), they are simply ignored in further processing steps (see step3 in Fig. 6.4).

Interpolation: Assigns values to the pixel grid of the virtual view by applying an arbitrary interpolation method to the rendered pixels have floating point coordinates.

Edge Smoothing: Applies a low pass filter of color values over a small area around all edges. The edges' positions are known by re-using positions from the previous edge detection and the subsequent warping. This smoothing counteracts the pixilation that occurs at the edges.

6.3 Proposed method for view interpolation

In the view interpolation case, where we have access to data from two input views, the method in Section 6.2 is extended by adding a merging technique. The proposed edge-aided rendering method for MVD is shown in Fig. 6.5. Excluding the blending technique, the processing steps are similar to the method applied for V+D in Section 6.2. Thus only merging step is presented here.

When two views are warped from the reference views into the virtual view point, the next step is to combine information from the different views, which is called merging or blending of views. In the edge-aided rendering approach, the *merging* step (see Fig. 6.5) combines the projected pixels from the two views by applying a weighted averaging of subsets. To do so, the two projected view coordinates are first sorted in ascending order. The pixel that is closer to the camera is selected by observing the depth information when two projected pixel coordinates are close enough (less than a threshold). In this step, the horizontal differences $\Delta x = x_i - x_{i-1}$ between two of the projected coordinates are calculated and compared with the threshold d_0. This step is applied irrespective of the pixels origin information (see Fig. 6.6(a)).

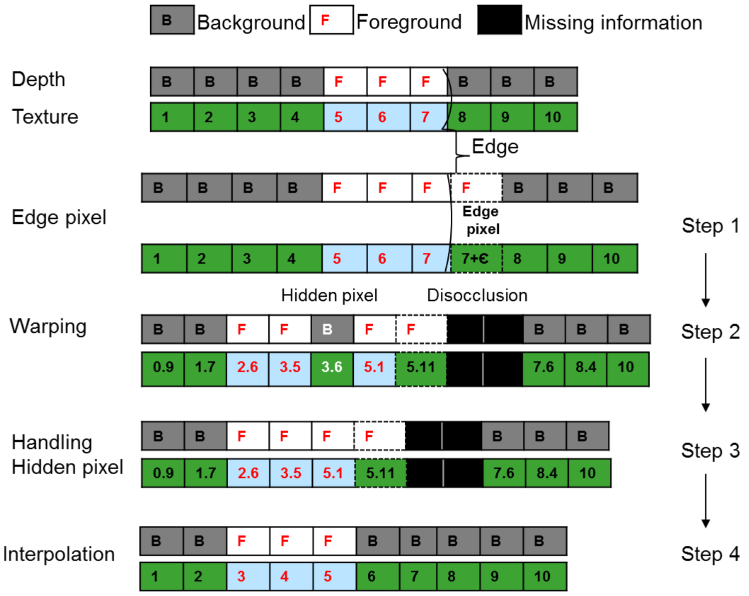


Figure 6.4: Edge-aided rendering method description with the steps edge-pixels, warping, handling hidden pixels and interpolation. Note that the numbers in the texture box indicates assumed pixel positions and the pixels positions changes with depth after warping.

After the previous step, the total image width is then divided into one-pixel-wide bins, i.e., the virtual view pixel position is $x \pm 0.5$. A weighted average is computed for all pixels that are projected in each of these bins, where the weight is based on the distance to each pixel's original image. An example of weighted average values of two view pixels within a bin are shown in Fig. 6.6(b) with $\otimes$. Next, all pixels within each bin are assigned this averaged value and then interpolation is applied (see interpolated values $+$ in Fig. 6.6(b)).

Causes related to the rendering artifacts and solutions of the proposed approach and VSRS are shown in Table 6.2.

6.4 Methodology

Quality assessments of rendered views are generally performed using quick objective measurements. However, objective metrics do not provide exact visual quality because they fail to include all aspects of the HVS. On the other hand, subjective tests using human subjects offer a good assessment but require time and careful planning to setup the tests. Thus we employed the commonly used objective metrics and visual comparison using expert researchers in this field to assess the quality of the rendered view. The common objective metrics used in the rendered view as-

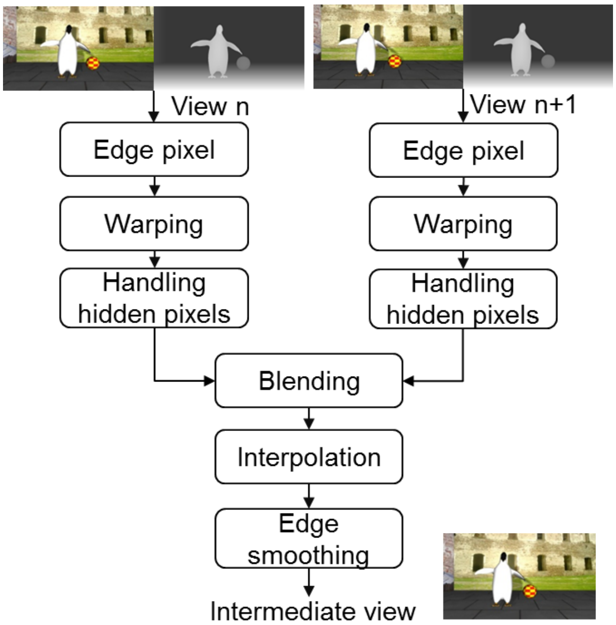
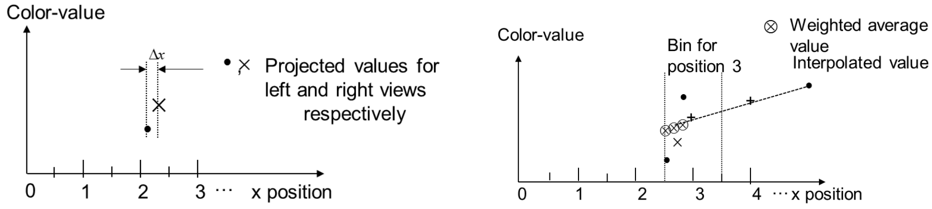


Figure 6.5: Edge-aided rendering method for view interpolation: Each V+D data is aided with edge pixels and warped to intermediate view position then two views data is merged with using proposed blending method and applies interpolation.



(a) Merging1

(b) Merging2

Figure 6.6: Different steps in merging.

sessment are Peak Signal-to-Noise-Ratio for the intensity component (YPSNR) and Mean Structural Similarity Index (MSSIM) [WBSS04]. Both metrics measure the view quality, where the MSSIM score corresponds better to the experienced visual quality. The higher value of YPSNR and the value of MSSIM that is closer to 1 better demonstrates the quality.

The input data used for the rendered view assessment in this dissertation are computer generated and photographic with various texture and depth characteristics. This data was used to validate the proposed view rendering methods in different conditions. Excluding the "Penguin", other sequences were captured in the real world. The sequences are “Newspaper” from the 3DVC reference set, “Ballet”, “Breakdancers”, “Lovebird1”, “Poznan Street” and “Poznan Hall” [ZKU$^+$04, UBH$^+$08, DGK$^+$09, NPS08]. Details of the sequence characteristics are presented in Table 6.1. The hole size in the table is given as follows: first the percentage of hole pixels in each image is computed; then the average value of the percentage of holes pixels over the sequence is calculated. Note that hole size is measured in view extrapolation case for the closest available camera view point. All of the test sequences are used in the V+D scenario, which means that the V+D input data is used to produce extrapolated virtual views. The virtual views are rendered to virtual camera positions so that they match available real camera views. This allows the extrapolated virtual views to be compared with the ground truth depth and texture, which is a pre-requisite for a full-reference objective metric evaluation. Moreover, the notation used for the rendered views, given as v2→v1, indicates that the original camera view 2 is warped to the virtual camera view 1.

6.4.1 View extrapolation

The proposed method was assessed by computing extrapolated virtual views and comparing them to captured views at the same position. Edge-pixels were implemented by adding an extra pixel with horizontal shift $\epsilon = \pm 0.01$, where the sign is given by the depth map derivative at the edge. We applied both nearest neighbor and linear interpolations. The low-pass edge smoothing filter was a combination of bicubic interpolation and an averaging filter.

Three MVD test sequences with various texture and depth characteristics were used to validate the proposed view extrapolation method. The test sequences are “Penguin”, “Lovebird1” and “Poznan Street”. The texture and depth characteristics of the test sequences are listed in Table 6.1. The photographic test data consisted of an extract of the first 10 frames of the MVD sequences “Lovebird1” and “Poznan Street”. The rendered views in the comparison are “Penguin” v2→v1, “Lovebird1” v6→v4 and “Poznan Street” v4→v5 respectively. The quality of the rendered views was measured using the metrics as presented in Section 6.4. The proposed method was tested on a frame-by-frame basis, without considering time effects. The quality assessment was performed using both objective measurements and visual inspection as presented in Section 6.4.

The view quality achieved by the proposed method was compared to the quality obtained by VSRS using 1D approach, i.e. line-by-line rendering. VSRS have incorporated many tools proposed in the literature to correct rendering artifacts (see Table 6.2).

6.4.2 View interpolation

In view interpolation, the proposed method was assessed by computing intermediate views and comparing them to the original views. Three input MVD test sequences were used for the assessment as test sources (SRC): SRC1: ‘Poznan Hall”,

Table 6.1: Test input data characteristics

Sequence Name	**Resolution**	**Camera arrangement**	**Texture background**	**Depth Properties**	**Hole size**(Avg)
Penguin	1280x720	3 cameras with 3.25 cm spacing, 1D arrangement	Medium structured	Large depth discontinuities and few layers	3%
Ballet	1024x768	8 cameras with 30 cm spacing, 1D arc arrangement	Low structured	Large depth discontinuities and many layers	13.5%
Breakdancers	1024x768	8 cameras with 20 cm spacing, 1D arc arrangement	Low structured	Small depth discontinuities and few layers	6.3%
Lovebird1	1024x768	12 cameras with 3.5 cm spacing, 1D arrangement	High structured	Large depth discontinuities and few layers	4.3%
Newspaper	1024x768	9 cameras with 5 cm spacing, 1D arrangement	Medium structured	Small depth discontinuities and many layers	10%
Poznan street	1920x1088	9 cameras with 13.5 cm spacing, 1D arrangement	High structured	Small depth discontinuities and many layers	5%
Poznan hall	1920x1088	9 cameras with 13.5 cm spacing, 1D arrangement	High structured	Small depth discontinuities and many layers	3.5%

SRC2: "Poznan Street" and SRC3: "Lovebird1" respectively. The sequence details are presented in Table 6.1. For each SRC, two intermediate views were rendered at $\lambda = 0.25$ and $\lambda = 0.5$ between the two original cameras, where λ is the interpolation parameter, $\lambda = 0$ corresponds to the left original view and $\lambda = 1$ corresponds to the right original view. The quality assessment was performed using both objective measurements and a subjective test as mentioned in Section 6.4. The first 10 frames

of the selected sequences were considered for the objective measurements and a few key frames were manually selected based on large disocclusions and the scene content for the subjective test.

The proposed method for view interpolation (M1) was compared to the following state-of-the-art methods: MPEG-VSRS 3.5 [vsr10] and fast 1-dimensional view synthesis algorithm software[Hhi12], which we denote as M2 and M3, respectively. As mentioned in Chapter 5, both these reference methods have incorporated many tools in the rendering process to remove the artifacts.

Objective evaluation

The evaluation metric used in the objective test setup was the MSSIM as this metric has shown good agreement with subjective tests. These objective measurements were performed at the $\lambda = 0.5$ view position since no reference images are available for view position $\lambda = 0.25$. In SRC1, view 6 was rendered from view 5 and view 7. For SRC2, view 4 was rendered from view 3 and view 5 and for SRC3, view 6 was rendered from view 4 and view 8.

Subjective evaluation

The subjective quality test procedure was chosen according to the goal of the study.

Test Procedure: The most commonly used test procedures from ITU-T Rec. P.910 are absolute categorical rating (ACR) and pair comparison (PC) [ITU08, ITU12]. In this experiment, PC subjective test methodology was utilized in order to obtain reliable quality ratings. PC is the suitable method when small differences exist between the images from the various test conditions.

Apparatus and environment: The test content was presented to the observers in monoscopic mode using Alienware display (Optx AW2210, 1920x1080 full HD LCD). The subjective assessment session was conducted according to the ITU test environment. This test included the following characteristics; viewing distance: three times display height, the peak luminance of the screen: 200 cd/m^2, the ratio of the luminance of the background behind the picture monitor to the luminance of picture: 0.15, chromacity of the background: D$_{65}$ and the background illumination was less than or equal to 20 lux [ITU08].

Test material and error conditions: The test sources were SRC1 (frame 150), SRC2 (frame 1) and SRC3 (frame 1) and hypothetical reference circuits (HRCs) were the proposed method M1 and reference methods M2 and M3 respectively.

Test subjects, training and randomization: A 16 naive test observers participated in the test. All of the subjects were engineering and science students who ranged in age from 20-35 years old. A pre-screening was conducted for all participants to check for the visual acuity and color blindness by using the Snellen chart and the Ishihara chart. A training session was conducted prior to the test to help the subjects understand the task. Four pairs of images that included two test image that had been rendered with different methods were paired and presented to the subjects in random order. Two test images with different rendering methods were paired and presented to the subjects in random order. The images were presented one by one and subjects were asked to choose one image from each pair as their preferred image. Subjects were free to toggle between images in each pair as many times as they liked before making their choice.

Analysis: Preferences from all subjects were then converted into a quality score using the Bradley-Terry's model. This model gives maximum likelihood estimators for scale parameters with confidence intervals, the hypothesis test for the model fit, uniformity and preferences among groups [Han01].

6.5 Results and analysis

6.5.1 View extrapolation

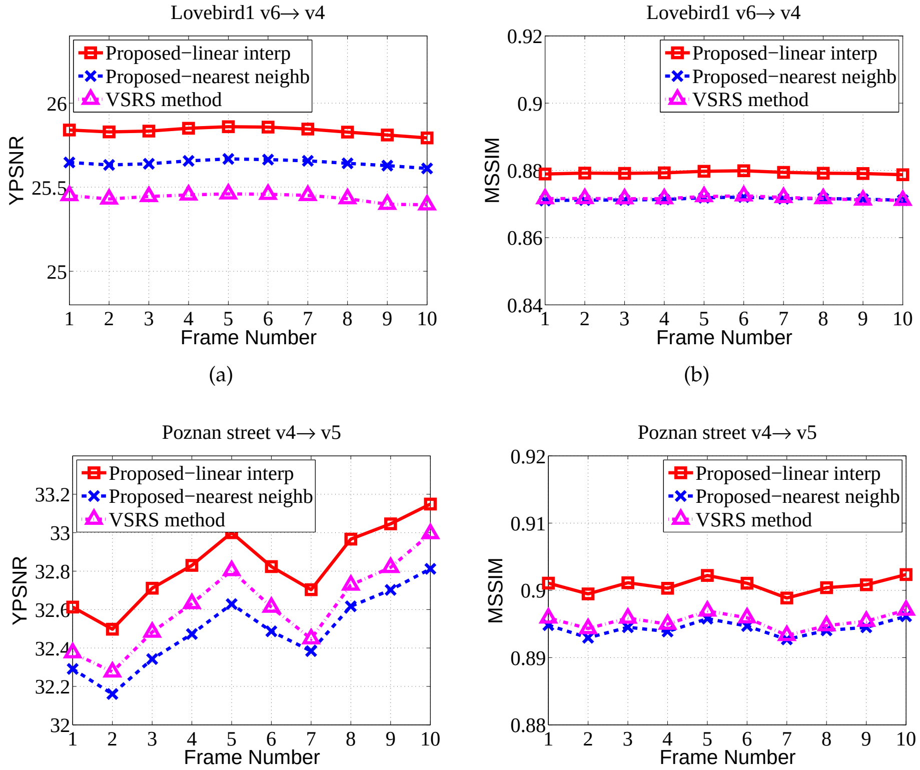
Major artifacts due to DIBR are summarized in Table 6.2 along with the applied remedies in VSRS and the proposed method.

Empty cracks are a consequence of assigning each warped pixel to integer coordinates in the virtual view. In most DIBR methods, these cracks are found and filled by median filtering. For the proposed method, these cracks do not appear because the floating-point coordinates of the rendered pixels are temporally stored and the values at integer coordinates are then interpolated. Therefore, interpolation is shown to be a simple way to avoid empty cracks.

Translucent cracks occur for the same reasons as empty cracks but contain the background color as there happens to be such information at that position. VSRS puts constraints on pixel mappings in order to remove translucent cracks. The proposed method removes all hidden pixels directly in the rendering process, thus no translucent cracks will appear in the resulting virtual view.

Holes regions are uncovered areas that resulted from the warping process. VSRS propagates the background to fill these regions. In the proposed method, the issue of the empty regions is resolved with the introduced edge-pixels and interpolation. In fact, the effect is equivalent to background propagation for empty regions.

Smearing of edges appears due to smooth texture edges or depth-texture misalignment. In DIBR using MVD sequences, this artifact corresponds to Corona-like effects. To avoid this artifact, VSRS removes original pixels that would create smearing and fills the possible empty region in the virtual view using the background propagation. The proposed method resolves this issue by selecting pure background and foreground color values for edge-pixels in the original view. The proposed method selects the color at one to two pixels from each side of the edge in the original view based on the fact that each pixel integrates all light rays over the pixel surface in the capturing process. Hence, one pixel value might be a combination of both background and foreground colors at the edges, incorporating more or less of each component depending on how much of each reaches that particular pixel. Therefore,



(a) (b)

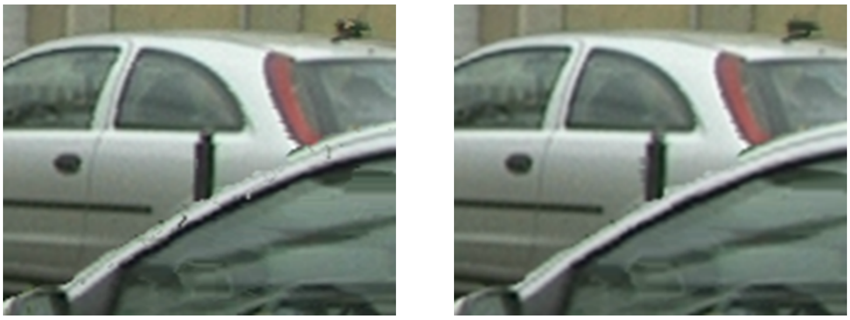
(c) (d)

Figure 6.7: View extrapolation objective evaluation: (a) YPSNR for each rendered frame “Lovebird1” (v6→v4); (b) MSSIM for each rendered frame “Lovebird1” (v6→v4); (c) YPSNR for each rendered frame “Poznan street” (v4→v5); (d)MSSIM for each rendered frame “Poznan street” (v4→v5);

the pure background and foreground color should in principle be at most one pixel away. Shading and other edge effects may however increase that distance.

Unnatural contours are the pixilation of “new” edges between the background and the foreground in the virtual view. They are the consequence of a sharp edge between the foreground and the background where pixel colors at the edge have not been blended. VSRS employs splatting along the edges in order to remove this jagged appearance. This process is the re-creation of averaged color values for edge pixels (explained above in the paragraph on smearing of edges).

The YPSNR and MSSIM graphs consistently demonstrate that the proposed method with linear interpolations performs better than the nearest neighbor interpolation and the VSRS method. In one case, the nearest neighbor interpolation resulted in a similar quality to the result of the VSRS, see Fig. 6.7. Fig. 6.8 (a) and (b) depict details of a rendered frame at view position 3. It exhibits the visual improvements



(a) (b)

Figure 6.8: View extrapolation visual quality results: "Poznan Street" (v4→v5) image :(a) VSRS; (b) Proposed method.

at the edges when employing the proposed method over VSRS. The obtained results using the computer generated image demonstrate negligible visual differences between the proposed method and VSRS.

6.5.2 View interpolation

The objective measurements using MSSIM are shown in Fig. 6.9 and the subjective test results from the pair comparisons can be seen in Fig. 6.10. The quality scores are presented for each SRC using the three rendering methods; M1, M2 and M3.

The MSSIM values show improved results for all three test sequences using the proposed method M1 compared to the other state-of-the-art methods; M2 and M3, see Fig. 6.9.

According to the subjective scores, the proposed method; M1 performed better than M3 at the rendered view position $\lambda = 0.5$, but at the other investigated view position; $\lambda = 0.25$, no significance difference could be noted between the results from the proposed method and those from the reference methods. The reason for this is that the rendered view is closer to the original view in the case of $\lambda = 0.25$ (see Fig. 6.10(a)). In Fig. 6.10(b), no significant difference can be observed between the results from the proposed method and those from the reference methods at either of the rendered view positions. This may be due to the few but distinct depth changes in the scene, because there is too little information that depends on the edge-aided pixels. Fig. 6.10(c) demonstrates that the proposed method M1 performs better than the reference method M3 at $\lambda = 0.5$, but not at other view positions. This is due to linear weighting for the nearest original view in the merging when the two input views have slightly different depth characteristics.

Results from the three SRCs reveal that the subjective scores from the tests depend on the test material as well as the rendered view position. The overall scores in Fig. 6.10(d) confirm that the results from the proposed method M1 are comparable to the state-of-the-art methods. The proposed method is, however, straightforward

Table 6.2: Comparison of rendering artifact solutions

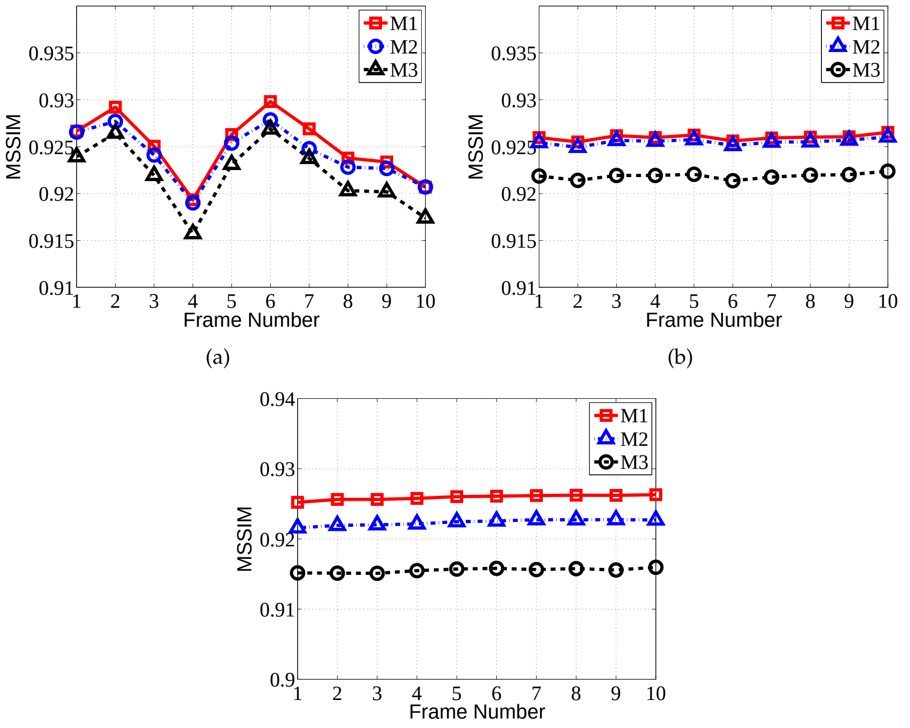
Artifacts	Cause	Remedy	
		VSRS	Proposed Method
Empty cracks	Integer round offs of projected coordinates	Median filtering	Interpolation
Translucent cracks	Background pixels seep through cracks	Constraints on pixel mapping order	Removal of hidden pixel
Holes	Uncovered areas	Background propagation	Introduction of edge pixels, interpolation
Smearing of edges	Smooth texture edges or depth texture misalignment	Removal of unreliable pixels before warping	Selection of "pure" background and foreground colors for edge pixels
Unnatural contours	Pixilation at edges due to new border between foreground and background	Splatting along edges	Upscale filtering along edges

and requires less dedicated processing for artifacts in the DIBR.

The objective evaluation shows that the proposed method improves quality to a certain extent, especially for sequences with a background that has low frequency texture. The subjective test results could not determine a significant difference between the proposed method and the state-of-the-art methods. Nonetheless, this result is encouraging because the proposed method employs a simple and straightforward processing structure, where the reference methods include specific processing steps to remove different artifacts.

6.6 Concluding remarks

This chapter presented the investigations of DIBR artifacts. Moreover, an alternative formulation of DIBR for view extrapolation using V+D has been presented. In particular, the proposed method relies on fundamental principles: forward warp-



(a)

(b)

(c)

Figure 6.9: View interpolation objective evaluation: Objective metric MSSIM of investigated images; (a) MSSIM for each rendered frame at view position 6 of "Poznan Hall"; (b) MSSIM for each rendered frame at view position 4 of "Poznan Street"; (c) MSSIM for each rendered frame at view position 6 of "Lovebird1".

ing and interpolation in the rendered view. The artifacts that were consequences of the 3D warping process were studied. Their remedies in the DIBR implementation VSRS and the proposed method were analyzed. It was shown that the proposed method reduces most inherent artifacts by introducing edge-pixels and by using the interpolation so that specific solutions are not required for each separate artifact. Furthermore, the proposed method demonstrated better quality when edges in the rendered view were smoothed by low-pass filtering.

An extension of this method for view interpolation using MVD has been presented. The projected pixels from adjacent original views were then merged using weighted averaging, followed by linear interpolation to give the values on the virtual view pixel grid. The objective evaluation showed a slightly improved quality for the rendered views using the proposed method. The subjective evaluation could not determine a significant difference from the state-of-the-art methods. Nonetheless, the results are encouraging because the proposed method omits specific processing steps for removing artifacts in the virtual view.

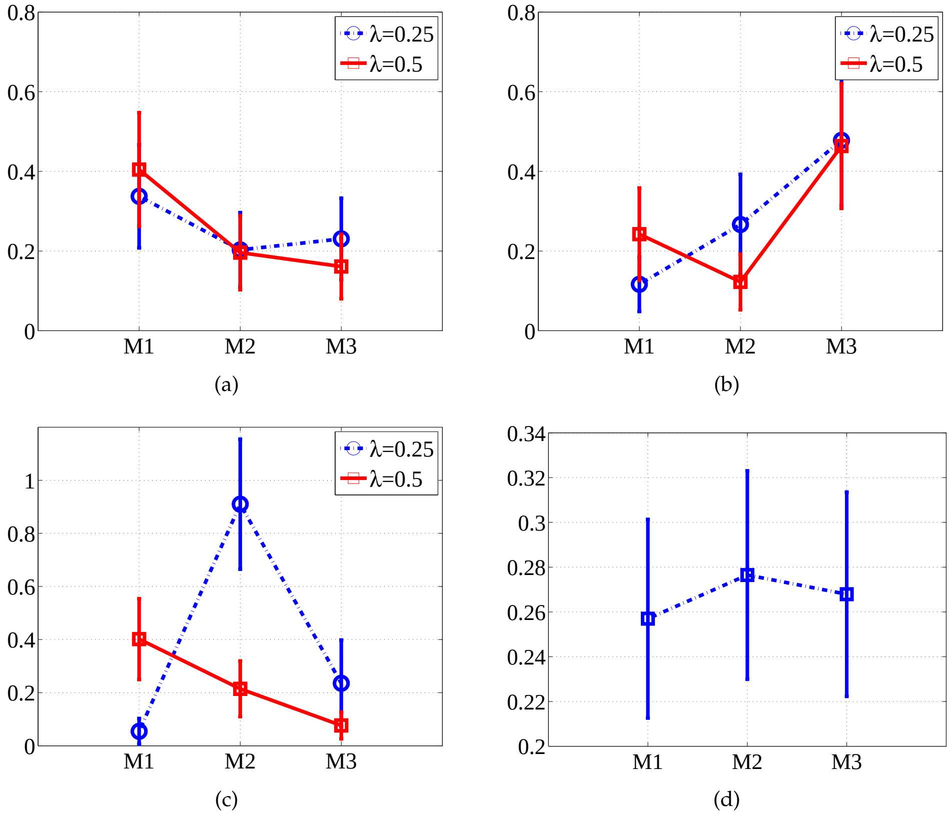


Figure 6.10: View interpolation subjective evaluation: Subjective quality score for each sequence at different view positions (λ is interpolation parameter $\lambda = 0$ corresponds to left original view and $\lambda = 1$ corresponds to right original view); (a) "Poznan Hall"; (b) 'Poznan Street"; (c) "Lovebird1"; (d) Overall score for each method.

6.6.1 Contributions

The author contributed to this chapter by providing an analysis of the cause of DIBR artifacts and by explaining the limitations of the reference methods. The author also provided an alternative DIBR method that included a straight-forward approach to reducing artifacts. The work in this chapter was presented in Papers I and II. Proposed edge-aided rendering method for view extrapolation was presented in [MSO12], and a subsequent work that examined view interpolation was presented in [MSOT13].

Chapter 7

Hole-Filling Using Depth-Based Inpainting

An edge-aided rendering method using interpolation was introduced in Chapter 6 to handle the holes in virtual views that are rendered using DIBR method. Interpolation method shows acceptable results for small base line setups when the holes are small or appear between smooth textures. However, the interpolation results appear unnatural and spatially inconsistent, when holes are larger. Thus, alternative solutions for handling holes are proposed in this chapter.

This chapter will address the holes by proposing a new inpainting method as an extension to the exemplar-based methods discussed in Chapter 6. The proposed method adds depth constraints to the exemplar-based method. The effect of depth in various steps of the inpainting process will also be discussed in this chapter. Moreover, a depth edge-based classification will be presented to reduce depth-based inpainting artifacts. The resulting improvements will be discussed at the end of Chapter 7.

7.1 Introduction

Disocclusions are one of the major problems in virtual views when rendered from DIBR. Disocclusion are holes occurring at depth discontinuities near the object borders. However, where they appear between the objects depends on the position of virtual view camera. There are mainly two cases where disocclusion can appear: (I) between the foreground and the background, and (II) between two different parts of the foreground. An example of case (I) is shown in Figure 7.1(a) between the woman's leg and the floor. Case (II) is exemplified in Figure 7.1(b), the area between the woman's head and hand. Disocclusions in the virtual view belong to background information. Thus, finding the background and filling the holes with background information is plausible.

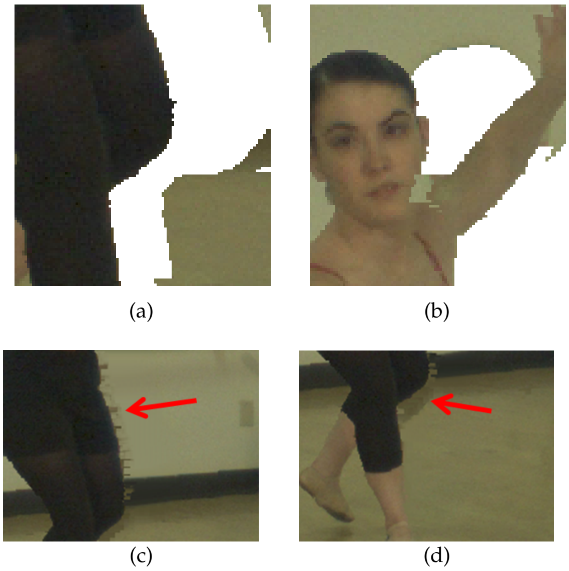


Figure 7.1: Holes in rendered image (holes in white color) and depth-based inpainting artifacts: (a) Holes between the foreground and the background; (b) Holes between two different parts of the foreground; (c) Jaggedness; (d) Background-leaking.

Chapter 5 presented various hole-filling techniques using diffusion-based and exemplar-based inpainting to fill holes in the virtual views. Prior depth-based inpainting methods that aimed to handle disocclusion with the background textures still possessed artifacts, i.e. *jaggedness* and *background-leaking*. *Jaggedness* is an inconsistent texture along the foreground object (see Fig. 7.1(c)). It is caused by restricting the patch-matching in the depth-based inpainting step to the background when a target patch contains both foreground and background pixels. The *background−leaking* is a propagation of inconsistent background data into different depth layers (see Fig. 7.1(d)). Background−leaking appears when the patch-matching is constrained to one depth layer, whereas the target patch contains multiple layers' depth values. Despite many efforts in the field, there is still a need for improvements which give better visual quality. Thus, this chapter will address the following questions in the context of hole-filling to improve the quality of virtual views. i) How to handle disocclusions in virtual views in order to improve the quality of the rendered view in a visually plausible manner? ii) How does the depth information influence the inpainted image quality? How to reduce depth-based inpainting artifacts?

To address the holes in the rendered view, a depth-based curvature inpainting is introduced by extending the Criminisi et al. inpainting method, which was described in Section 4.3. The novelties of the proposed method are introducing depth constraints in the boundary extraction, the data term and in the patch matching steps in order to fill the holes from background information. Furthermore, using the knowledge about the effect of depth in various steps of the inpainting process a depth edge-based classification in the patch matching is proposed in order to reduce depth-based inpainting artifacts.

The rest of this chapter is organized as follows: Sections 7.2, Section 7.3 explain the proposed depth-based inpainting and depth edge-based source region classification. The evaluation methodology is described in Section 7.4. The results and analysis are presented in Section 7.5, and finally, Section 7.6 concludes this chapter.

7.2 Proposed method

Fig. 7.2 illustrates the view rendering method by incorporating the proposed depth-based curvature inpainting module. Presented in Fig. 7.2, the texture and depth map are initially warped to the virtual view position and then pre-processing is applied. In the pre-processing step, artifacts such as cracks and ghosting are removed using interpolation and ghosting pixels are removed by extending the hole area on the background side. It is important to handle these artifacts prior to the hole-filling process to avoid the hole-filling process being affected by these artifacts.

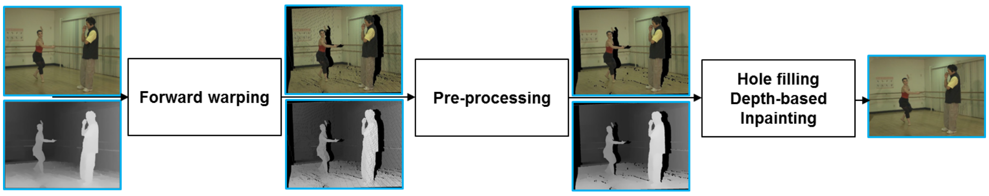


Figure 7.2: Block diagram of the rendering method using inpainting for view extrapolation.

After adding the depth information in the Criminisi et al. method, the steps of the depth-based curvature inpainting become:

A. Depth-based one-sided boundary

B. Depth-based curvature data term

C. Depth-based source region classification

Fig. 7.3 illustrates how these steps relate to the general inpainting process as described in Section 4.3. Step A consists of defining a depth-based one-sided boundary, which helps the filling process to start from the background. This step is related to the boundary extraction block in Fig. 7.3. In Step B, the Curvature Driven Diffusion (CDD) model [CS01] is used similar to [LWXD12], as a data term D in the priority computation: the data term is extended by incorporating the depth to give the importance to the isophote curvature and strength. This step is related to data term in the priority block in Fig. 7.3. Finally, in Step C, depth constraints are derived from the warped depth to avoid the foreground data from the source region and to favor the background filling. This step is related to source region classification and block matching in the patch matching and filling block in Fig. 7.3. Details of these steps are presented in the following sections.

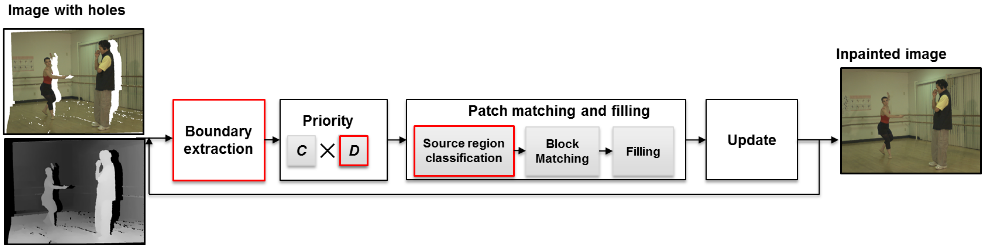
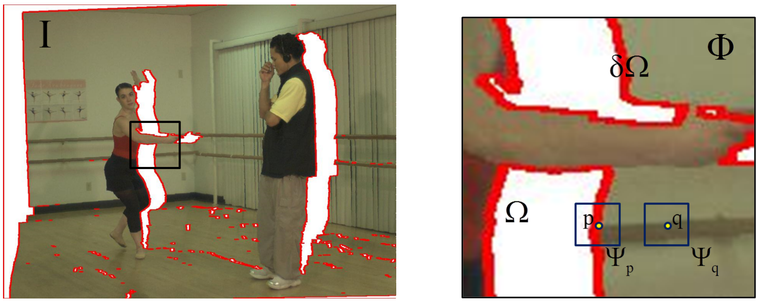


Figure 7.3: Block diagram of the exemplar-inpainting method; the highlighted blocks indicates the depth-based curvature inpainting method enhancements.



(a) (b)

Figure 7.4: Exemplar-based inpainting notation (holes in white color): (a) Warped texture image; (b) Notation diagram.

7.2.1 Depth-based one-sided boundary

Let $\mathbf{I}$ be input image with hole Ω, and the remaining portion of the image is defined as the source region $\Phi = \mathbf{I} - \Omega$, which is illustrated in Fig. 7.4. The background side of the hole boundary is obtained as follows: First, a one-sided boundary $\delta\Omega_1$ of the hole boundary is obtained by applying the convolution operation on a hole mask ($\mathbf{H}$) as given by (7.1). Second, the directional priority selection is constrained to avoid foreground boundary when the holes appear between two foreground objects by using a depth threshold on $\delta\Omega_1$, such that pixels whose depth values are less than μ percent of the maximum depth value in the warped depth map are selected.

$$\delta\Omega_1 = \mathbf{H} * \mathrm{E} \tag{7.1}$$

$$\mathbf{H}(\mathbf{q}) = \begin{cases} 1 & \text{if } \mathbf{q} \in \Omega; \\ 0 & \text{otherwise.} \end{cases} \tag{7.2}$$

$$\delta\Omega_1^{'} = \delta\Omega_1 \ \forall \mathbf{p} : \mathbf{Z}(\mathbf{p}) < \mu \cdot max(\mathbf{Z}), \tag{7.3}$$

where $\delta\Omega_1^{'}$ is the depth-based one-sided boundary, $\mathbf{Z}$ is the depth map and $\mathbf{Z}(\mathbf{p})$ is the depth value at pixel location $\mathbf{p}$. The convolution kernel E is defined as follows depending on the warped view:

$$\mathrm{E} = \begin{cases} \begin{bmatrix} 1 & -1 & 0 \end{bmatrix} & \text{if left warped view ;} \\ \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} & \text{if right warped view.} \end{cases} \tag{7.4}$$

Once the hole boundary is obtained, using (7.3), priority term $P = C \cdot D$ is calculated using confidence term that was given in (4.4) with

$$C(\mathbf{p}) = \frac{1}{|\Psi_{\mathbf{p}}|} \sum_{\mathbf{q} \in \Psi_{\mathbf{p}} \cap \Omega} C(\mathbf{q}), \tag{7.5}$$

$$C(\mathbf{q}) = \begin{cases} 0 & \text{if } \mathbf{q} \in \Omega; \\ 1 & \text{otherwise.} \end{cases} \tag{7.6}$$

and the data term (7.7). Thereafter, the holes in the background region are filled using the selected depth guided direction priority. However, filling with the depth-based one-sided boundary can handle holes to a certain depth level, and the remaining holes are filled with one-sided boundary priority. Moreover, when the virtual camera setup is not horizontal, the holes appear depending on the view angle of the virtual view camera. Then the hole filling is processed with the full boundary extraction by applying the convolution operation on a hole mask with Laplacian kernel. The full boundary, one-sided boundary and the depth-based one-sided boundaries are shown in Fig. 7.5.

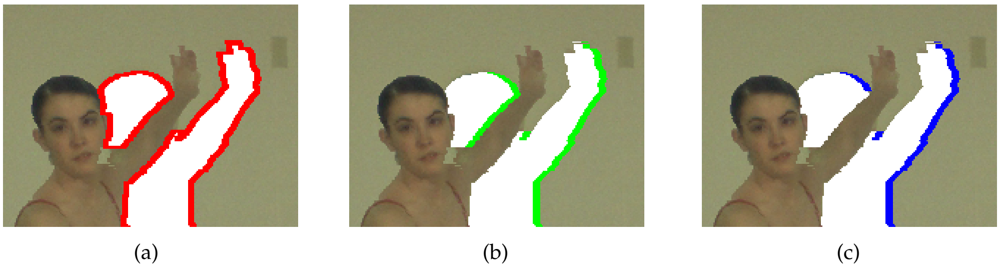


Figure 7.5: Boundary extraction: (a) Full hole boundary; (b) One-sided hole boundary; (b) Depth-based one-sided hole boundary;

7.2.2 Depth-based curvature data term

The curvature driven diffusion (CDD) data term in [LWXD12] is used as the data term in the proposed inpainting method. This is because the adopted CDD model in (4.3) aids the proposed method with finding structure using curvature information. The CDD model in (4.3) fills the discontinuity parts by considering strength of the isophote and curvature, which is the geometry of the isophote. In addition, the proposed inpainting method is extended with the depth information, as we introduce the depth curvature along with the texture in the data term, to compute the structure information. Hence the data term is given by:

$$D\left(\mathbf{p}\right) = \left| \nabla \cdot \left(\frac{\kappa_{\mathbf{p}}}{|\nabla \mathbf{I_p}|} \nabla \mathbf{I_p} \right) \right| , \tag{7.7}$$

$$\kappa_{\mathbf{p}} = \nabla \cdot \left(\frac{\nabla \mathbf{I_p}}{|\nabla \mathbf{I_p}|} \right) , \tag{7.8}$$

where $\kappa_{\mathbf{p}}$ is the curvature of the isophote through a pixel $\mathbf{p}$. The data term is calculated using both texture and depth gradients.

7.2.3 Depth-based source region classification

Unlike methods [DS11] and [GMG11], the proposed method employs the source region classification using warped depth information, in order to select similar patches from the nearest depth range. The source region is divided into foreground-background regions using depth information in the target patch. Considering Φ to be the known source region, which contains both the foreground and the background regions, the best source patch selection from the foreground region, can be avoided by sub-dividing Φ using depth threshold T_{U} according to:

$$\Phi_{\mathrm{B}} = \Phi - \Phi_{\mathrm{F}}, \tag{7.9}$$

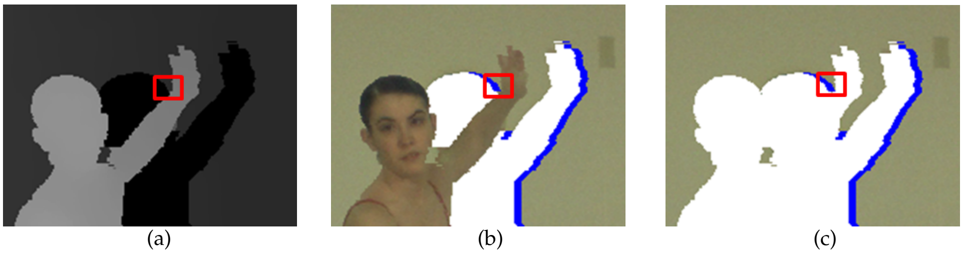


Figure 7.6: Depth-based source region classification: (a) Warped depth image; (b) Source region; (c) Depth-based source region;

where Φ_{F} is the foreground source region whose depth values are higher than the depth threshold T_{U}.

The depth threshold has two different values selected adaptively from the variance of the known pixel values of the target depth patch. When the depth patch lies near the foreground (See Fig. 7.6(a)), the variance of the target depth patch is greater than a threshold γ, and the patch might contain unwanted foreground values (See Fig. 7.6(b)). The average value of the depth patch is then chosen instead as the depth threshold in order to deduct the foreground parts. Otherwise, the patch contains the uniform or continuous depth values, so the maximum value in the depth patch is used as the depth threshold in order to get the best patch according to the depth level. The depth threshold T_{U} is defined as follows:

$$T_{\mathrm{U}} = \begin{cases} \overline{\mathbf{Z}_{\hat{\mathbf{p}}}} & \text{if } \mathrm{var}(\mathbf{Z}_{\hat{\mathbf{p}}}(\mathbf{q})|_{\mathbf{q}\in(\Psi_{\hat{\mathbf{p}}}\cap\Phi)}) > \gamma; \\ \max(\mathbf{Z}_{\hat{\mathbf{p}}}) & \text{otherwise.} \end{cases} \tag{7.10}$$

$\Psi_{\hat{\mathbf{p}}}$ is the target patch, $\mathbf{Z}_{\hat{\mathbf{p}}}$ is the depth patch centered at $\hat{\mathbf{p}}$; and $\overline{\mathbf{Z}_{\hat{\mathbf{p}}}}$ is the average value of the depth patch. $\mathbf{Z}_{\hat{\mathbf{p}}}(\mathbf{q})$ is the depth value at pixel $\mathbf{q}$, var is the variance and γ is the depth variance threshold.

Once the target patch $\Psi_{\hat{\mathbf{p}}}$ from the priority term and the depth-based source region Φ_{B} defined in (7.9) are computed, the target patch is filled with the best N_{b} number of patches within the source region.

$$\Psi_{\mathbf{q}} = \arg \min_{\Psi_{\mathbf{q}}\in\Phi_{\mathrm{B}}} \left\{ \|\Psi_{\hat{\mathbf{p}}} - \Psi_{\mathbf{q}}\|_2^2 + \beta \cdot \|\mathbf{Z}_{\hat{\mathbf{p}}} - \mathbf{Z}_{\mathbf{q}}\|_2^2) \right\}, \tag{7.11}$$

where, $\Psi_{\hat{\mathbf{p}}}$, $Z_{\hat{\mathbf{p}}}$ are target texture and depth patches and $\Psi_{\mathbf{q}}$ is source patch. $\|\Psi_{\hat{\mathbf{p}}} - \Psi_{\mathbf{q}}\|_2^2$ is the sum of squared difference between known pixels in two patches $\Psi_{\hat{\mathbf{p}}}$, $\Psi_{\mathbf{q}}$ and β is a weighting coefficient to equalize the effect of the depth and texture. In contrast to the prior methods, the proposed method used the warped depth information in inpainting. In order to help the inpainting using the depth information, the holes in the depth image should be filled simultaneously, or the hole free depth image should be available for depth-based inpainting process. In the proposed method, we considered that the depth map should be filled simultaneously along with the texture.

Similar to [GMG11] and inspired by [WSI07], we used a weighted average of N_b patches from the patch-matching step to fill the missing information in the disocclusion area. Weighted averaging minimizes the noise in the selected patch and helps the smooth continuation of the filling process. Once the best patches are obtained, holes in the depth map are filled by copying the depth values from the best patch location. Holes in the texture image are filled with a weighted average of N_b patches of texture values.

The data term is calculated iteratively after every target patch is filled, because the new copied texture to fill the hole region is a combination of the N_b best patches. In addition, the source region is updated such that the filled area is also available as source region for the next target patch.

7.3 Proposed depth edge-based source region classification

To address the depth-based inpainting artifacts that are presented in Section 7.1, a depth edge-based source region classification is proposed by extending the classification in Section 7.2.3. The artifacts are due to the foreground pixels in the target patch during block matching and depth classification using data from the target patch. To reduce these artifacts, we propose excluding the foreground pixels from both block matching and from the target patch in the patch matching step of inpainting. The method must therefore identify the foreground pixels in the target patch, which is carried out by finding the appropriate thresholds related to the depth. The method can be described in two steps: (1) Foreground identification at patch level and (2) Determine a depth threshold for background selection. The steps are related to the patch matching and the filling process (see Figure 7.3). In general, the depth values on the borders of a disocclusion area are similar to the depth values at the depth discontinuity in the original view (see Figure 7.7). Therefore, for Step (1), the depth distribution of patches on both sides of a hole is used in the warped depth image, and the patch at the depth discontinuity in the original depth image. In Step (2), a depth threshold is selected by using the data from Step (1). The threshold is used to classify the source region and the target patch so that only the background information is used in the filling of the disocclusion area.

7.3.1 Foreground identification at patch level

The depth values from the warped and original depth are compared in order to identify the foreground values. Assume that $\mathbf{Z_p}$ is a target depth patch that is centered at pixel $\mathbf{p}$ in the warped depth map, and $\mathbf{f}$ is a foreground pixel that relates to $\mathbf{p}$ with its original corresponding pixels $\mathbf{f}_o$ and $\mathbf{p}_o$ being neighbors. Depth patches that

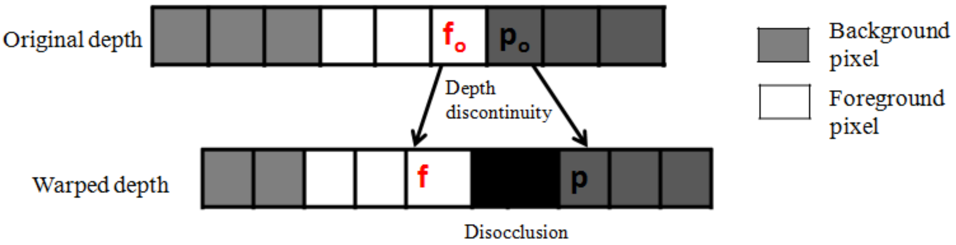


Figure 7.7: Illustration of original and warped pixel locations.

are centered at foreground pixels in original and warped depths are named $\mathbf{Z_{f_o}}$ and $\mathbf{Z_f}$ respectively (see Figure 7.8(a) and (b)). The known depth values of $\mathbf{Z_p}$ and $\mathbf{Z_f}$ are combined into a set $A = \mathbf{Z_f} \cup \mathbf{Z_p}$, which can be used later to determine depth distribution.

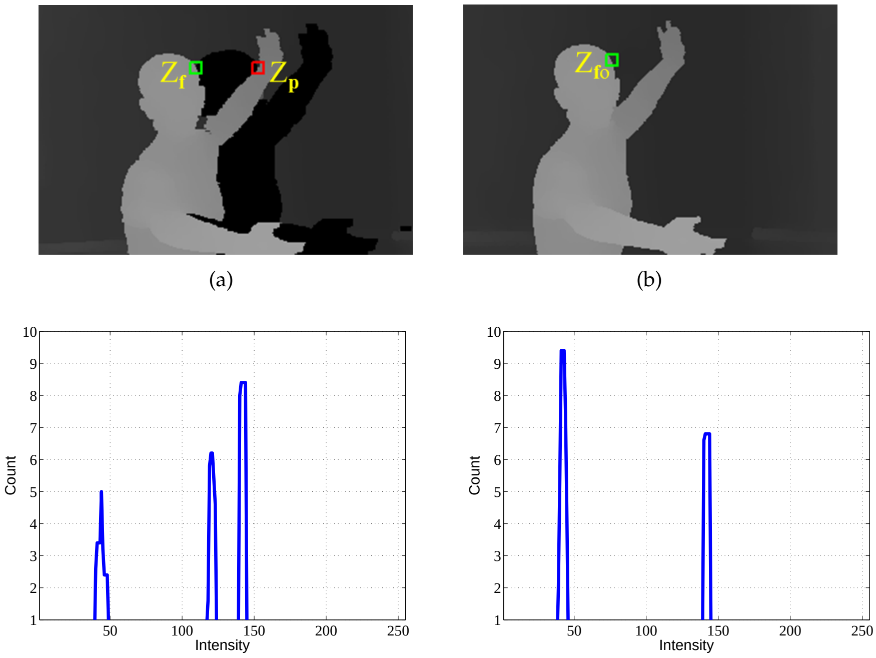
In case (I) where holes between the foreground and the background, both the target and the foreground patch share common information when the target patch $\mathbf{Z_p}$ is positioned on a foreground object. Patches that contain foreground pixels are identified by comparing $\mathbf{Z_p}$ and $\mathbf{Z_f}$ according to (7.12). In case (II) where holes between two different parts of the foreground, the foreground patch $\mathbf{Z_f}$ has no relation with the target patch $\mathbf{Z_p}$, because the depth values of $\mathbf{Z_f}$ differ from those in the target patch, but they do not correspond to the background. Foreground pixels are identified by comparing the depth values in A and the depth values in $\mathbf{Z_{f_o}}$ according to (7.12). Similarity of the depth distributions are measured by using the difference of local maxima positions in the histograms of A and $\mathbf{Z_{f_o}}$ (see Figure 7.8(c) and (d)). As this measure only works when the number of the local maxima is equal in the two histograms, the difference in the standard deviation is included as a measure for the spread in the given data. As a result, patches that contain foreground pixels are identified using the depth patch average and the histogram data.

$$\mathbf{Z_p} \in \begin{cases} \mathrm{F}_1 & \text{if } \overline{\mathbf{Z_f}} \leq \max(\mathbf{Z_p}); \\ \mathrm{F}_2 & \text{if } (T_1 \neq T_2) \wedge (\Delta\sigma > \xi), \end{cases} \tag{7.12}$$

where F_1 and F_2 are foreground labels for two different depth ranges in a target patch in case (I) and case (II) respectively. $\overline{\mathbf{Z_f}}$ is the average value of the depth patch $\mathbf{Z_f}$. T_1 and T_2 denote the difference of local maxima positions in the histograms of A and $\mathbf{Z_{f_o}}$ respectively. $\Delta\sigma$ is the difference between the standard deviation of A and $\mathbf{Z_{f_o}}$. ξ is the threshold to identify the foreground object pixels that are unrelated to the disocclusion, and that are calculated by computing the average of $\Delta\sigma$ along one side of the disocclusions' boundary with respect to the virtual view.

7.3.2 Depth-threshold for background selection

After the foreground pixels in the target patch are identified, a depth threshold is selected using local depth values in order to segment the source region into the fore-



(a) (b)

(c) (d)

Figure 7.8: Depth distributions in warped and original image patch: (a) Warped depth; (b) Original depth; (c) Histogram of the patch in warped image; (c) Histograms of the patch in original image(d).

ground and the background. The depth threshold has two different values depending on the depth values in the target patch. If the target patch contains foreground pixels, an average of depth values in the target patch distinguishes the foreground and the background. In this case, the average of the local target patch is used as the threshold value for preventing the foreground. In the case that the target patch contains background information and the source region has a gradient in depth, the depth values in the target patch are not sufficient to find the required background data. This problem is solved by selecting the threshold between the two local maxima in the histogram of the original depth patch, noting that the original depth patch at depth discontinuities contains at least two local maxima. In general, depth might have a number of layers, therefore the threshold is selected by taking the average of the depth values at the last two local maxima positions in the original depth patch histogram. This is because the last local maxima corresponds to the foreground. The depth threshold T_{U} is defined as:

$$T_{\mathrm{U}} = \begin{cases} \overline{\mathbf{Z_p}} & \text{if } \mathbf{Z_p} \in \mathrm{F}_1 \cup \mathrm{F}_2; \\ \frac{d_1+d_2}{2} & \text{otherwise,} \end{cases} \tag{7.13}$$

where $\overline{\mathbf{Z_p}}$ is the average of known pixels in the target patch and d_1 and d_2 are the last two local maxima depth values in the histogram of $\mathbf{Z}_{\mathbf{f}_o}$. After the best-source patches are found, only background pixels are copied to the missing region in the target patch.

7.4 Methodology

The impact of the depth-based curvature inpainting method on the rendering was evaluated on rendered views similar to the setup in Section 6.4 by using objective measurements as well as visual comparison. A set of 10 frames were selected from the following three MVD sequences "Ballet", "Break dancers" and "Lovebird1" for objective evaluation purposes. The sequence details were presented in Table 6.1 in the previous chapter. The test sequences have different depth and texture characteristics, which make them suitable for testing different disocclusion filling attributes of inpainting methods as the method relies on depth. The rendered views used for comparison are "Ballet" v5→v4, "Break dancers" v5→v4 and "Lovebird1" v6→v4 respectively. The virtual views were first rendered and small holes and ghosting artifacts were removed using a pre-processing step (see Fig. 7.2). Thereafter, the processed warped views were used as inputs for the inpainting method presented in Section 7.2. Important parameters of the proposed inpainting method were the patch matching window length of 120 pixels, $\mu = 0.4$ in (7.3), $\gamma = 80$ in (7.10), $\beta = 3$ in (7.11) and $N_\mathrm{b} = 5$.

The experimental setup was divided into three sections in order to evaluate the inpainting quality, influence of depth in the inpainting and depth classification in the patch matching as follows:

A. Comparison to related work

B. Sensitivity analysis

C. Depth edge-based source region classification

The details of the each setup are elaborated in the following three subsections.

7.4.1 Comparison to related work

In this test, the proposed method results were compared with reference inpainting methods [CPT04, DS11, GMG11]. It is worth noting that in the proposed method, the best exemplars were searched in the warped depth and texture images, where as in [DS11] and [GMG11] methods, exemplars were searched in the warped texture and original depth at the virtual camera position.

7.4.2 Sensitivity analysis

The setup of the sensitivity analysis was to identify the effects of the depth in the inpainting process. Thus sensitivity of the depth at various steps of the proposed inpainting process was analyzed by restricting the depth at those steps. The depth information is used in three steps of the proposed inpainting process:

- **Sensitivity to depth in boundary extraction**
  Sensitivity to the depth in boundary extraction (SZB) is analyzed by performing the inpainting method without considering the depth information, i.e., using only a one-sided boundary.
- **Sensitivity to depth in data term**
  Sensitivity to the depth in data term (SZD) is analyzed by computing the data term using R, G, and B channels without incorporating the depth information as an additional channel.
- **Sensitivity to depth in patch matching**
  The depth is used in source region classification and in block matching. Sensitivity to the depth in patch matching (SZP) is analyzed separately without using the depth. The sensitivity to the depth in source region classification (SZSR) and sensitivity to the depth in block matching (SZBM) are analyzed by using Φ as a source region and searching only in R, G, and B channels.

Note that the depth was not changed in other steps while measuring the sensitivity at one step.

7.4.3 Depth edge-based source region classification

In this experiment, the results from the following methods were compared: the state-of-the art reference methods [DS11, GMG11], proposed inpainting method using the depth edge-based source region classification and the proposed inpainting method using source region classification in Section 7.2.3.

7.5 Results and analysis

7.5.1 Comparison to related work

The objective evaluation results from the related work and proposed methods are shown in Fig. 7.10. The YPSNR and MSSIM graphs consistently demonstrate that the proposed depth-based curvature inpainting method performs better than the reference Criminisi et al., Daribo et al. and Gautier et al. methods. In addition to the objective results and for further visual assessments, Fig. 7.11 and Fig. 7.12 show synthesized views of the "Ballet" and "Lovebird1" images with the missing areas and inpainted images from different inpainting methods. Missing regions in Fig. 7.11(b) and Fig. 7.12(b) are filled with the foreground information since no information about the depth is used to assist the filling process. Although the Daribo et al. and Gautier et al. methods are aided with the true depth information, the missing areas are filled with the unwanted textures due to lack of depth constraints and the filling order, see Fig. 7.11(c),(d) and Fig. 7.12(c),(d).

The depth-based curvature inpainting method operates in a more realistic setting which depends on the warped depth only. The depth-based curvature inpainting method shows visual improvements compared to the reference methods. The results from Fig. 7.11(e) and Fig. 7.12(e) show that the depth-based curvature inpainting method propagates the necessary neighboring information into the missing areas, by retaining both smooth areas (at the left side of the "Ballet" image) and propagating neighborhood structure (on the curtain in the "Ballet" image and at the head of the women in the "Lovebird1" image). The inpainting method might not reproduce exact structures as in ground truth images due to the lack of knowledge about the scene contents, however it closely replicates the main structure.

7.5.2 Sensitivity analysis

The sensitivity to the depth in various steps of the depth-based curvature inpainting method is analyzed by using both the objective metrics and visual comparison. The results of the objective measurements are presented in Table 7.1 and 7.2. Fig. 7.13 and Fig. 7.14 show the synthesized views of the "Ballet" and "Lovebird1" images with the missing areas and sensitivity of the depth in depth-based curvature inpainting method for visual assessment. Average YPSNR and MSSIM values and the visual comparison consistently demonstrate that the influence of the depth depends on the scene content and available depth information.

Sensitivity to depth in boundary extraction step

The results from the depth sensitivity in boundary extraction show that the depth information is important for handling the disocclusions, plausibly when they occur between foreground objects (see Fig. 7.13(b)). Although depth information is used in source region classification and block matching in order to fill holes from the background, the holes are still filled with the foreground due to the lack of knowledge about the depth on the boundary. As a result, the foreground boundary is selected in the boundary extraction. Thus, adding the depth constraint on the selection of the boundary improves the hole filling process.

Sensitivity to depth in data term

The results from the sensitivity to the depth in the data term show that the depth information is less important for filling holes in the inpainted view (see Fig. 7.13(c) and Fig. 7.14(c)). However, the depth information gives priority to structures when the depth contains several layers. For example Fig. 7.9(a) contains several depth layers, whereas the other depth image contains fewer depth layers (see Fig. 7.9(b)). Moreover, the depth characteristics depend on the depth acquisition method. Thus, if there are some layers in the depth map enriching the depth information, it benefits the inpainting process.

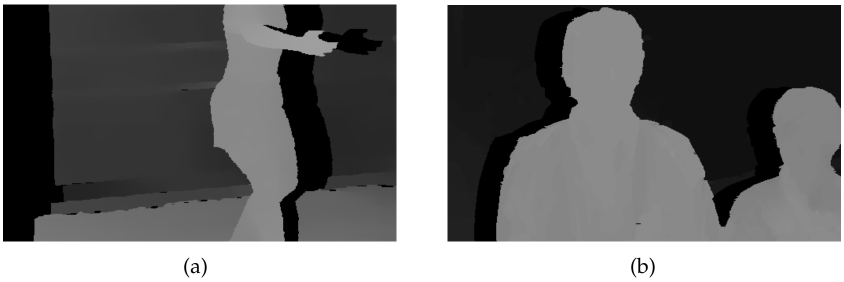


Figure 7.9: Different depth distributions in the background: (a) "Ballet" depth map; (b) "Lovebird1" depth map;

Sensitivity to depth in patch matching

The results from the depth sensitivity analysis in source region classification (SZSR) in the patch matching step demonstrate that the depth information is necessary in order to avoid the selection of similar texture regions from the foreground. Without using the depth information, regions between two foreground objects are filled with the foreground texture (see Fig. 7.13(d) and Fig. 7.14(d)). Other results from the sensitivity to the depth in source region classification (SZBM) in the patch matching step demonstrate that the use of depth in the source region is not as important when the depth data contains no layers (see Fig. 7.13(e) and Fig. 7.14(e)). In contrast, the depth data is essential for filling holes when the scene contains multiple depth layers in order to propagate the similar texture according to the depth level

In summary, the quality of the virtual view is highly dependent on the available depth information. Moreover, the depth-information plays a crucial role in filling the missing regions in the synthesized views by guiding the filling process to proceed from the background direction and copying the best texture from the background data. It is important that the depth map be filled with the background information. Otherwise, errors will propagate because all steps of the proposed inpainting method depend on the depth information.

7.5.3 Depth edge-based source region classification

Objective quality assessment results for the depth edge-based source region classification and patch matching and reference methods are listed in Table 7.3. The results show that the depth edge-based source region classification performs slightly better

Table 7.1: Sensitivity analysis: Averaged YPSNR.

Test sequence	SZB	SZD	SZSR	SZBM	Proposed
Ballet	31.91	31.88	31.76	**31.99**	31.96
Break Dancer	**31.84**	31.82	31.82	31.80	31.80
Love bird1	**25.16**	25.15	24.91	25.13	**25.16**

Table 7.2: Sensitivity analysis: Averaged MSSIM.

Test sequence	SZB	SZD	SZSR	SZBM	Proposed
Ballet	0.8745	0.8745	0.8741	**0.8748**	0.8742
Break Dancer	0.8289	0.8289	**0.8298**	0.8296	0.8289
Love bird1	0.8614	0.8613	0.8607	**0.8615**	**0.8615**

than the reference methods for the “Ballet” and “Breakdancers” sequences and gives similar results for the “Lovebird1” sequence. MSSIM values for the image subset (see Figure 7.15) is also presented in Table 7.4. The results show clear improvements in the results from the proposed method compared to the reference methods.

In addition to the objective results, inpainted images are visually compared (examples are shown in Figure 7.15). The visual comparison consistently demonstrates the superior performance of the proposed method compared to the reference methods. The proposed method especially outperforms other methods when considering the quality at the foreground object boundaries and thus produces visually pleasing results (see Figure 7.15(e)), whereas reference methods and source region classification using only the data from target patch and without removing foreground pixels, show artifacts around the object boundaries (see Figure 7.15(b) to (d)).

7.6 Concluding remarks

A depth-based curvature inpainting method for filling holes in the disocclusions has been presented in this chapter. The inpainted process is guided by the depth information in the filling direction, structure identification and in the patch matching. The results of the proposed depth-based inpainting method has been compared with the state-of-the-art reference methods using objective quality metrics and vi-

Table 7.3: Averaged MSSIM for the whole image.

Sequence	Daribo et al.	Gautier et al.	Proposed classification 7.2.3	Proposed classification 7.3
Ballet	0.860	0.865	0.874	**0.875**
Breakdancers	0.822	0.827	0.828	**0.829**
Love bird1	0.857	0.860	**0.861**	**0.861**

Table 7.4: MSSIM for the image subset shown in Figure 7.15(b) to (e)

Image	Daribo et al.	Gautier et al.	Proposed classification 7.2.3	Proposed classification 7.3
Ballet	0.709	0.807	0.814	**0.840**
Ballet	0.826	0.848	0.846	**0.861**
Breakdancers	0.769	0.774	0.829	**0.841**

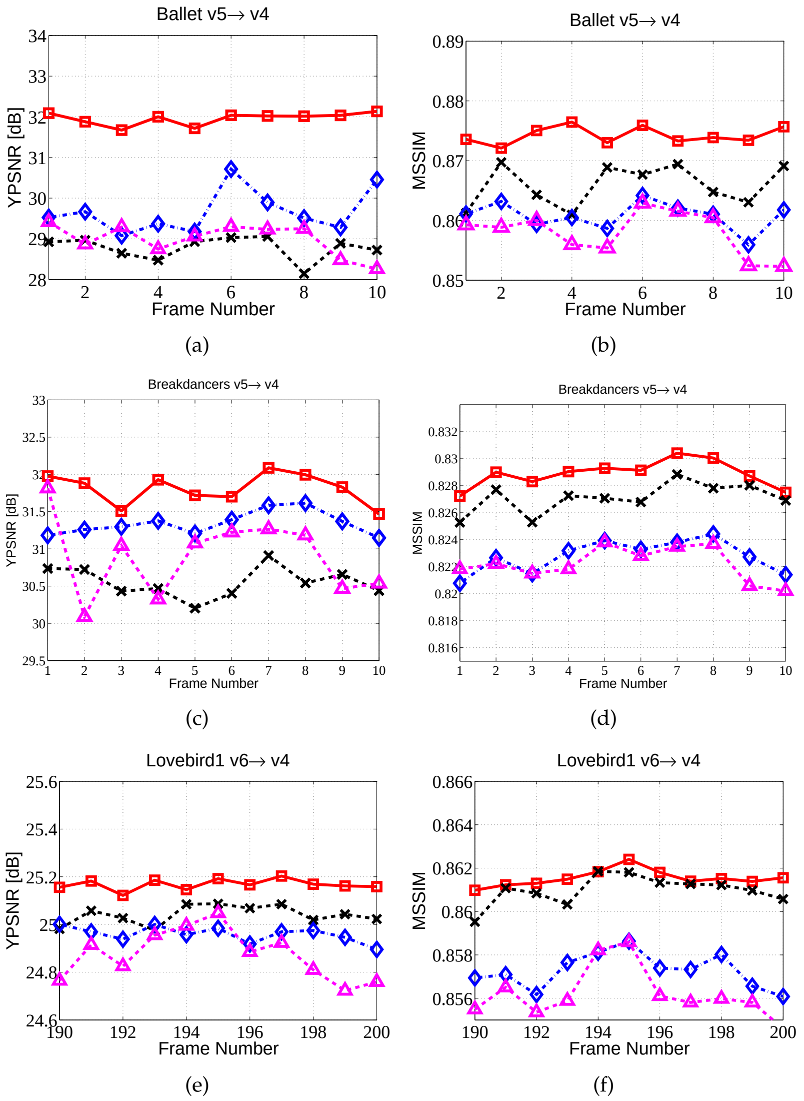
sual inspection. Both the objective and visual results consistently demonstrate that the proposed method offers an improved quality.

Furthermore, the influence of the depth information at each step of the inpainting process is analyzed by using objective measurements and visual inspection and the results are compared with the depth-based curvature inpainting method results. The evaluation demonstrated that to what degree the depth can be used in each step of the inpainting process depends on the depth distribution, which is presented in our visual analysis and objective evaluation. More elaborate knowledge about the depth distribution allows for tradeoffs that may reduce computational requirements without sacrificing quality. One such example is in the case where the scene has no disocclusions between the foreground and background and depth may be excluded from the boundary extraction step. Moreover, when the depth map contains only one foreground and background layer, the depth information in the data term and block matching demonstrate less impact on visual quality.

An extension of the inpainting method with an emphasis on patch matching has been presented. The method performs foreground and background classification locally using original and warped depth in order to fill the holes with background information according to the depth range. The proposed method excludes the foreground pixels from both the source region and the target patch during the patch matching process and thus boundary artifacts are removed. Experimental results have demonstrated improved objective quality and better visual quality.

7.6.1 Contributions

The contribution of the author to this chapter is providing improved hole-filling in the rendered views, analysing the impact of depth in various parts of the inpainting, and depth edge-based source region classification in the inpainting. The work in this chapter has been presented in separate publications. The proposed inpainting method was presented in Paper III [MOS13b], the analysis of depth in the inpainting process was presented in Paper IV [MOS13a] and finally the depth edge-based source region classification was presented in the Paper V [MSO14].

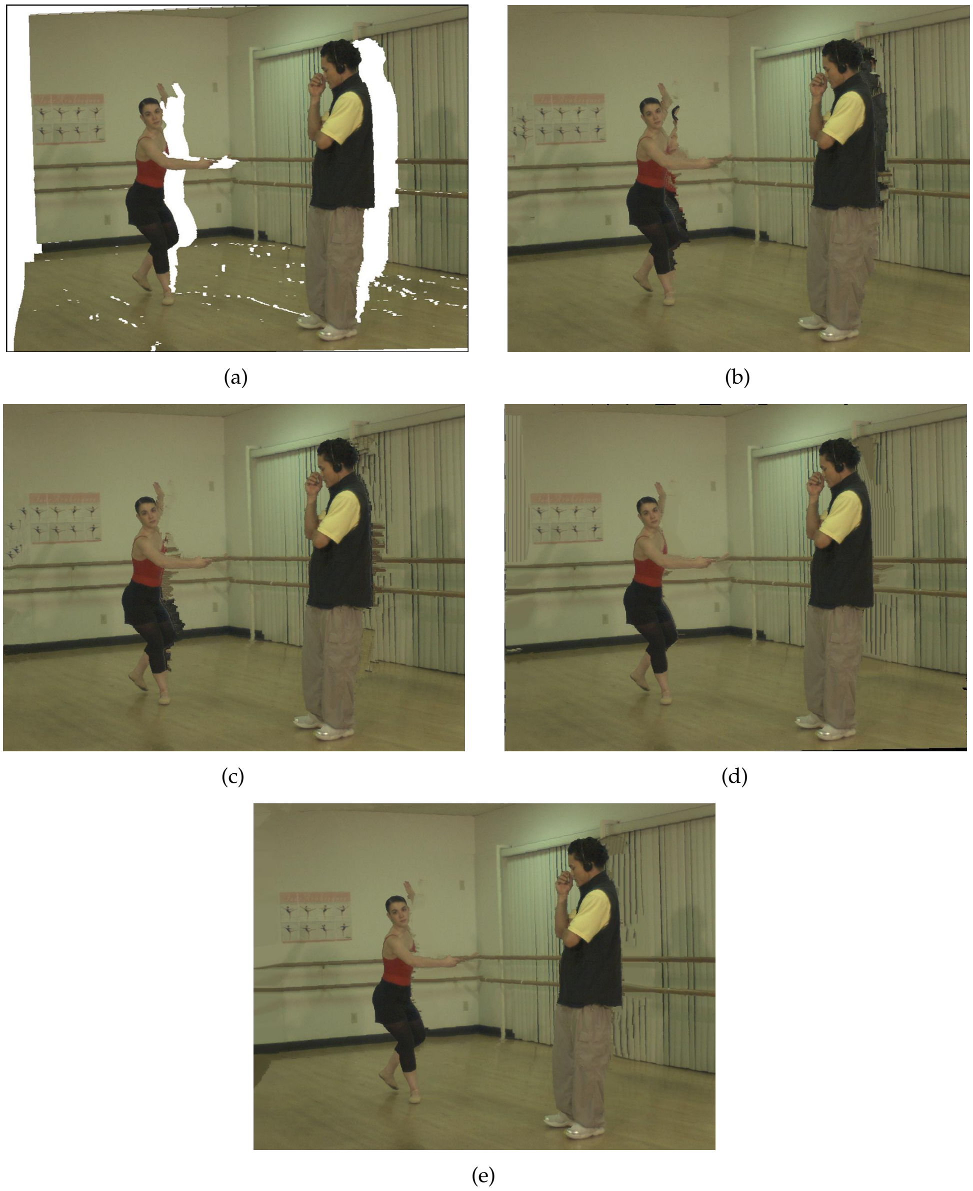


(a) (b)

(c) (d)

(e) (f)

Figure 7.10: Comparison to related work: Objective metrics YPSNR and MSSIM of the investigated sequences; Proposed method (−□−), Gautier et al. method (− × −) [GMG11], Daribo et al. method (− · ◊ · −) [DS11] and Criminisi et al. method (· · △ · ·) [CPT04]: "Ballet" (v5→v4) in (a), (b); "Break dancers" (v5→v4) in (c), (d); "Lovebird1" (v6→v4) in (e), (f).



(a)

(b)

(c)

(d)

(e)

Figure 7.11: Comparison to related work: inpainting method results of “Ballet” (v5→v4) frame1: (a) Warped image with holes marked; (b) Criminisi et al. [CPT04]; (c) Daribo et al. [DS11]; (d) Gautier et al. [GMG11]; (e) Proposed method using classification 7.2.3

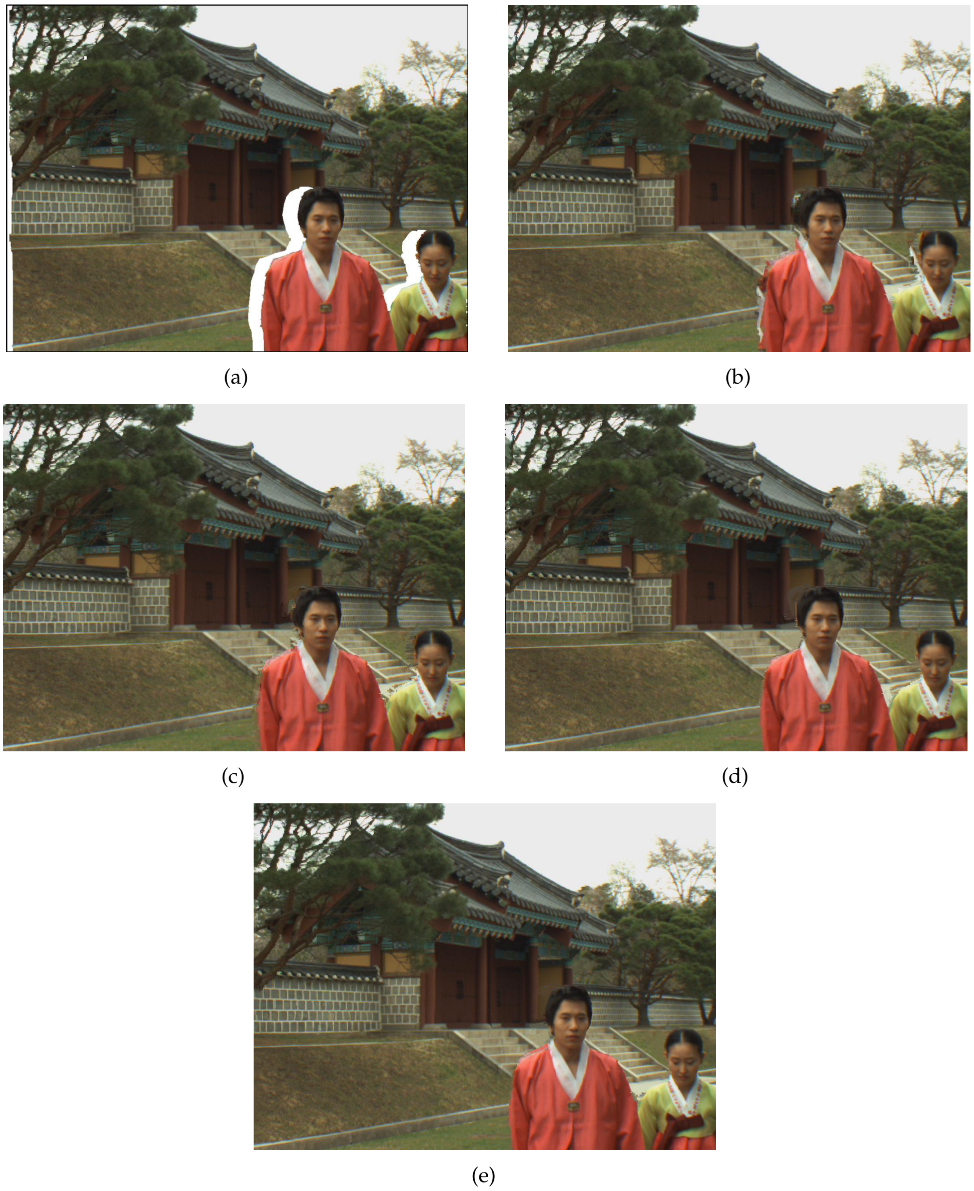
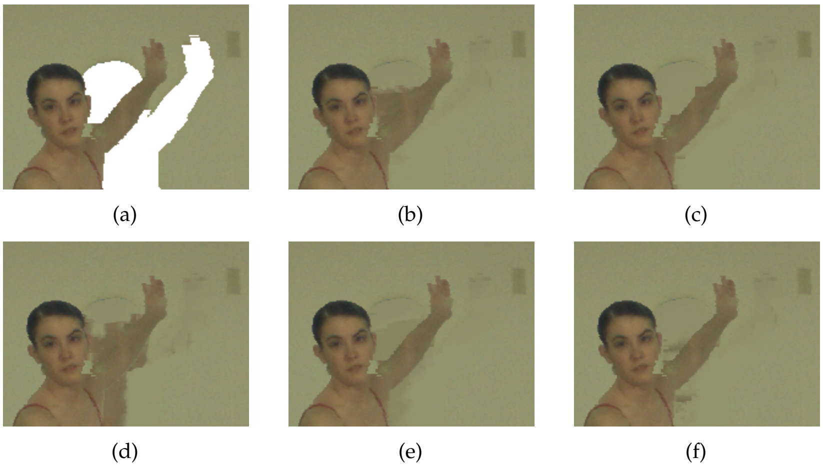
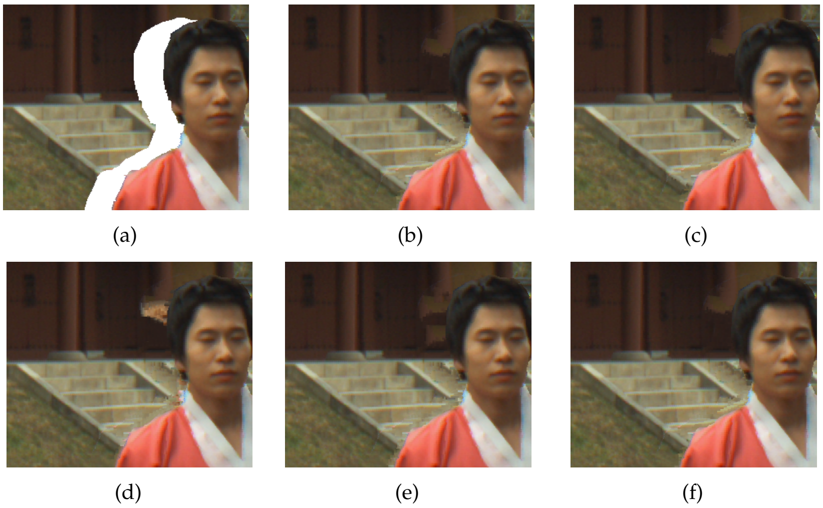


Figure 7.12: Comparison to related work: inpainting method results of “Lovebird1” (v6→v4) frame 190 : (a) Warped image with holes marked; (b) Criminisi et al. [CPT04]; (c) Daribo et al. [DS11]; (d) Gautier et al. [GMG11]; (e) Proposed method using classification 7.2.3.


(a) (b) (c)

(d) (e) (f)

Figure 7.13: Sensitivity analysis results for the investigated “Ballet” sequence frame 1: (a) Warped image (holes in white color); (b) Sensitivity to depth in boundary extraction (SZB); (c) Sensitivity to depth in data term (SZD); (d) Sensitivity to depth in source region classification (SZSR); (e) Sensitivity to depth in block matching (SZBM); (f) Depth in all steps.


(a) (b) (c)

(d) (e) (f)

Figure 7.14: Sensitivity analysis results for the investigated “Lovebird1” sequence frame198 : (a) Warped image (holes in white color); (b) Sensitivity to depth in boundary extraction (SZB); (c) Sensitivity to depth in data term (SZD); (d) Sensitivity to depth in source region classification (SZSR); (e) Sensitivity to depth in block matching (SZBM); (f) Depth in all steps.

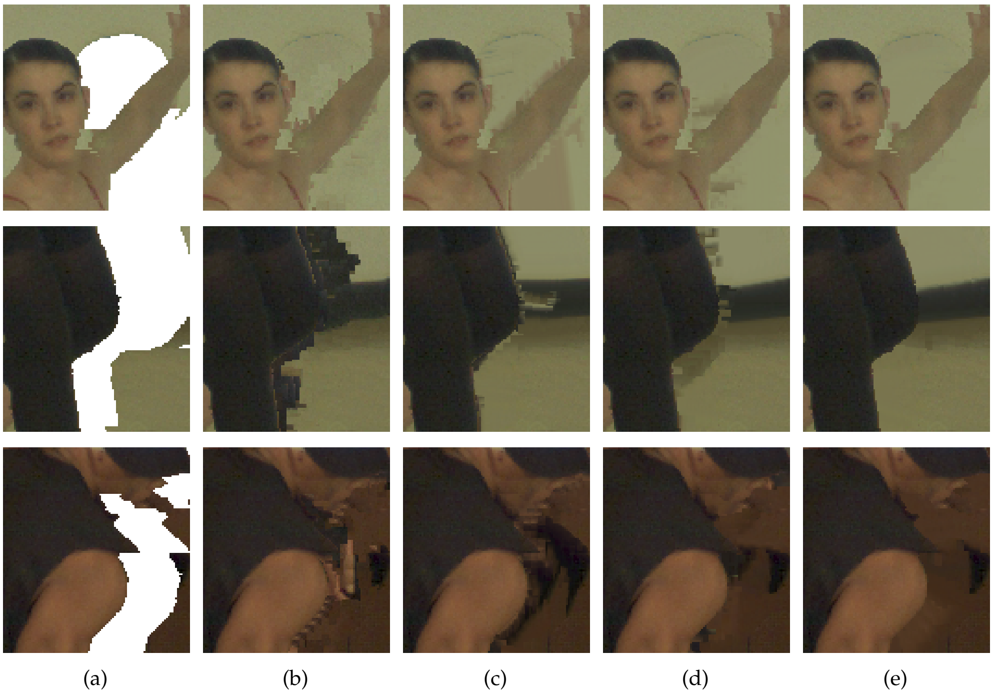


Figure 7.15: Depth edge-based source region classification: (a) Warped images; (b) Daribo et al. method [DS11]; (c) Gautier et al. method [GMG11], (d) Proposed method using classification 7.2.3; (e) Proposed method using classification 7.3.

Chapter 8

Rendering Using Layered Depth Image and Inpainting

In the previous chapter, the hole-filling method that uses the depth-based inpainting method was presented. This method effectively replicates the texture and structure at the disocclusions. However, filling holes in the virtual view might not provide the desired virtual view quality, since the virtual view already contains artifacts. Therefore alternative ways of rendering virtual views and hole-filling in the original view will be presented in this chapter.

This chapter will present a layered depth image (LDI) based approach to render views, where the LDI data are produced from a single V+D using the inpainting method described in Chapter 7. The created occlusion layers in the LDI correspond to disocclusions in the virtual view. Therefore disocclusions will not appear in the virtual view. Moreover, translucent problems can be reduced with the LDI formulation by identifying several occlusion layers.

8.1 Introduction

Inpainting in the virtual view relies on the background that is inconsistent due to the DIBR. Moreover, virtual views suffer from translucent disocclusion problems as presented in Section 3.4. These problems lead to spatial inconsistencies in the virtual view, see Fig. 8.1. Given the properties of DIBR, disocclusions in the virtual view are result of the regions that are occluded in the original view by the foreground as defined in Section 3.4. An illustration of virtual view generation is shown in Fig. 8.2, in which the occluded areas are highlighted with red and blue colors for the left and right virtual camera positions, respectively. Thus, the following question arises: what are the alternative ways to render views such that views are spatially consistent without artifacts?

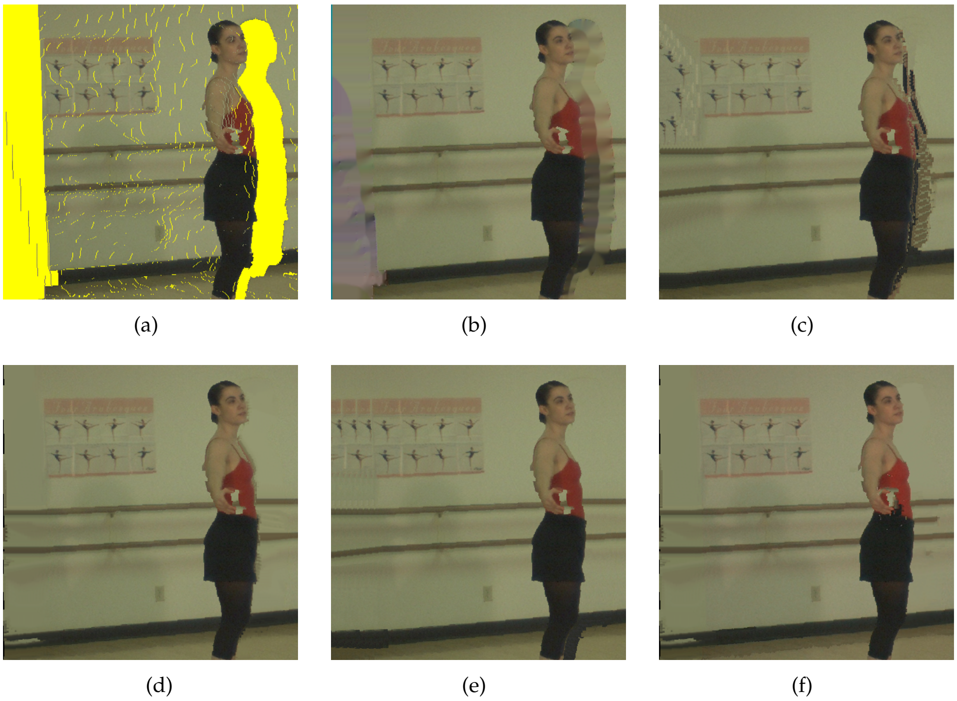


Figure 8.1: View synthesis results for an image of the sequence "Ballet" (frame28): (a) Warped image with holes (in yellow color); (b) VSRS method [vsr10]; (c) Daribo et al. method [DS11]; (d) Gautier et al. method [GMG11]; (e) Ahn et al. method [AK13]; (f) Wolinski et al. method [WMG13];

To improve spatial consistency in the virtual view we propose an alternative formulation of the view rendering. The proposed method has two main novelties:

1. Generating occlusion layers in the original view such that they correspond to the disocclusions in the virtual view.

2. Achieving spatially consistent inpainting results, which are facilitated by considering depth classification.

The LDI is generated starting from V+D data, where occlusion layers are inpainted with data from the original view. The occlusion layers are created in such a way that when the LDI is used to render a new virtual view, no disocclusions appear. Instead, view consistent data are produced, regardless of the virtual view position.

The work is limited to images. Moreover, the scope of this study does not encompass an optimization of all processing parameters. This chapter is organized as follows: Section 8.2 explains the proposed view synthesis method and depth-based inpainting. The methodology for evaluation described in Section 8.3. The results and analysis are presented in Section 8.4. Finally, Section 8.5 concludes this chapter.

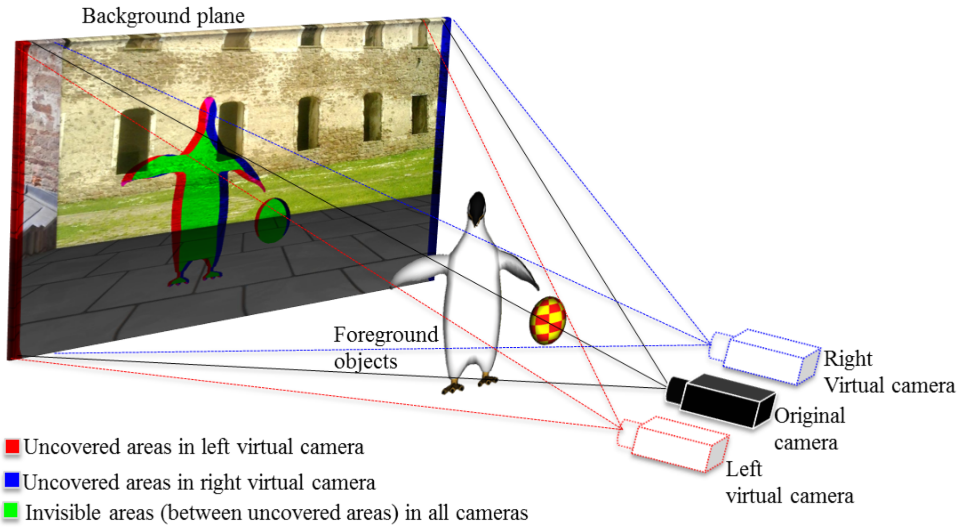

Background plane
Foreground objects
Right Virtual camera
Original camera
Left virtual camera
🟥 Uncovered areas in left virtual camera
🟦 Uncovered areas in right virtual camera
🟩 Invisible areas (between uncovered areas) in all cameras

Figure 8.2: Illustration of view-extrapolation of virtual cameras from original camera. Uncovered areas invisible in original camera become visible in the virtual cameras, so they need to be filled with content.

8.2 Proposed method

The proposed view synthesis method for extrapolating the virtual views is shown in Fig. 8.3. The main idea of the separate blocks are as follows:

3D warping produces a virtual view by employing DIBR. The projected pixels' position helps to locate the occlusion in the original view.

Occlusion mask and depth thresholds generation. The occlusion mask indicates where the occlusion data should be generated. Depth thresholds guides the

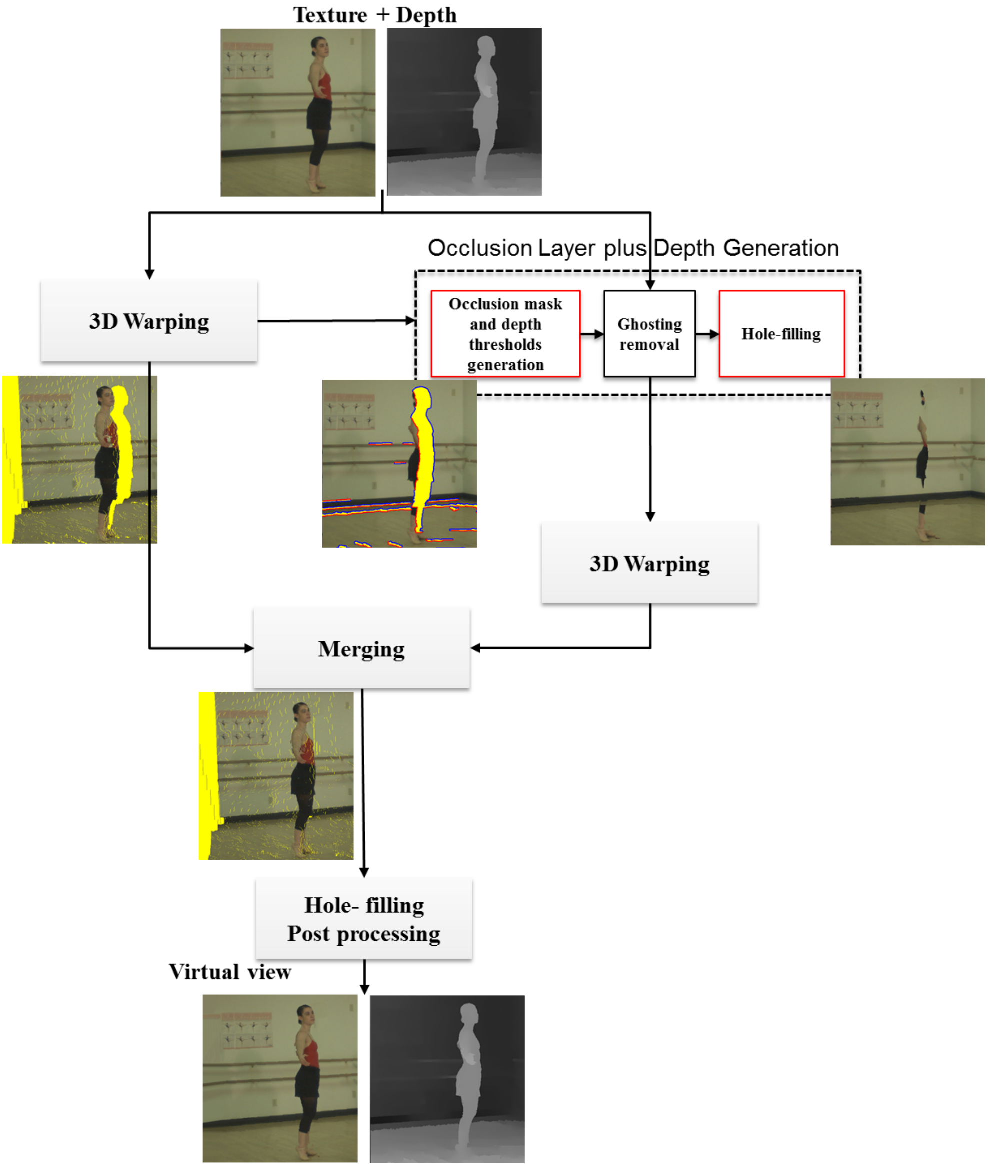

Texture + Depth
Occlusion Layer plus Depth Generation
3D Warping
Occlusion mask and depth thresholds generation
Ghosting removal
Hole-filling
3D Warping
Merging
Hole- filling Post processing
Virtual view

Figure 8.3: Proposed view rendering method using LDI generation and depth-based inpainting.

occlusion inpainting such that it achieves consistent background information.

Ghosting removal removes information near object borders and makes them into holes for later filling. These pixels consist of information blended from the foreground and the background that may cause ghosting if left untouched.

Hole-filling produces consistent background at the identified occlusions in the original view.

Merging combines the warped image with the warped occlusion layers, by which the disocclusions are avoided.

Hole-filling Post-processing classifies any remaining hole in the virtual view, caused by forward warping. These holes are manifested as cracks and out-of-field areas, and are filled by inpainting.

Details of the parts above are presented in the following sections, excluding 3D warping that was presented in Section 3.3.

8.2.1 Occlusion mask and depth thresholds generation

Estimating occlusions is a key part when producing LDI. The basic idea for locating the occlusions is based on the difference in projected pixel positions, similar to the DIBR methods in [BBG08, JGM11]. We use a depth threshold to locate the occlusions that appear at depth discontinuities .

Occlusion mask generation

Disocclusions occur between pixels that are neighboring in the original image, have a distinct difference in depth and that become separated in the warped image as shown in Fig. 8.4. Using this information we identify occlusions by first finding these pixels.

Let Γ be the warping operation that projects the original image $\mathbf{I}_\text{o}$ to virtual view image $\mathbf{I}_\text{v}$. A depth discontinuity pixel (DDP) is a pixel pair $\{\mathbf{f}_\text{o}, \mathbf{p}_\text{o}\}$ positioned as horizontal neighbors in the original view and that satisfy the following conditions:

$$\text{DDP} = \{\mathbf{f}_\text{o}, \mathbf{p}_\text{o}\}\,;\, \|\mathbf{d}_\text{v}\|_2 > \eta, \tag{8.1}$$

where η is an occlusion threshold to identify occlusion, and $\mathbf{d}_\text{v}$ is a displacement vector is given by

$$\mathbf{d}_\text{v} = \begin{pmatrix} d_x \\ d_y \end{pmatrix} = \mathbf{f}_\text{v} - \mathbf{p}_\text{v}, \tag{8.2}$$

$$\begin{aligned} \Gamma :& \mathbf{f}_\text{o} \rightarrow \mathbf{f}_\text{v}, \\ \Gamma :& \mathbf{p}_\text{o} \rightarrow \mathbf{p}_\text{v}, \end{aligned} \tag{8.3}$$

where $\mathbf{f}_\text{v}$, $\mathbf{p}_\text{v}$ are projected pixels in $\mathbf{I}_\text{v}$.

Next, the pixels in the DDP are labeled as foreground and background pixels using their respective depth values. When there are multiple depth layers, DDPs might overlap and produce inconsistent foreground-background labeling, meaning

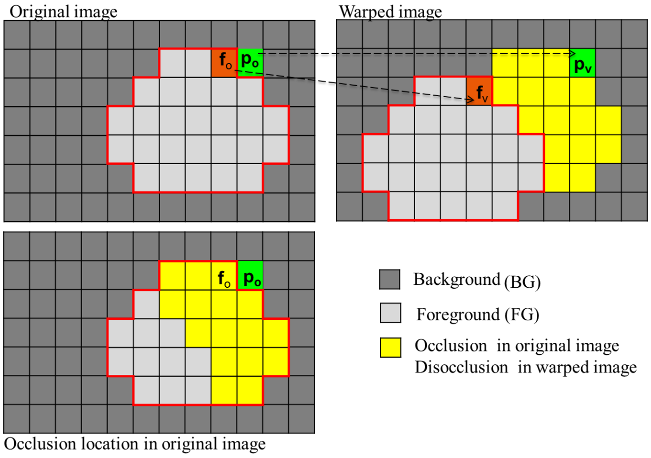

Original image
Warped image
f_o
p_o
p_v
f_v
f_o
p_o
Background (BG)
Foreground (FG)
Occlusion in original image
Disocclusion in warped image
Occlusion location in original image

Figure 8.4: Identification of occlusion location

that a pixel labeled as background in one DDP would be labeled foreground in another DDP. To circumvent this issue, a labeling refinement step is applied that selects either the background or foreground based on the difference in projected pixels' position.

Once the DDPs are identified using (8.1) and subsequently labeled, the occlusion is located by finding the foreground using a depth threshold and displacement vector, since the occlusions are caused by the foreground. Thereafter, the pixels that belong to the occlusion as identified at DDP $\{\mathbf{p}_\mathrm{o}, \mathbf{f}_\mathrm{o}\}$ is given by:

$$\mathrm{H}\left(\mathbf{q}\right) = \begin{cases} 1 & \text{if } (\mathrm{Z}(\mathbf{q}) > T_1), \forall \mathbf{q} \in \overline{\mathbf{p}_\mathrm{o}\mathbf{r}_\mathrm{o}}; \\ 0 & \text{otherwise.} \end{cases} \tag{8.4}$$

where $\overline{\mathbf{p}_\mathrm{o}\mathbf{r}_\mathrm{o}}$ is a line between pixels $\mathbf{p}_\mathrm{o}$ and $\mathbf{r}_\mathrm{o} = \mathbf{p}_\mathrm{o} - \mathbf{d}_\mathrm{v}$, H is an occlusion mask, and T_1 is an occlusion depth threshold, which is the average of the depth discontinuity pixels depth values:

$$T_1 = \frac{\mathrm{Z}(\mathbf{f}_\mathrm{o}) + \mathrm{Z}(\mathbf{p}_\mathrm{o})}{2}, \tag{8.5}$$

pixels with depth values larger than the depth threshold T_1 are foreground. Occlusions for all the DDPs can be located using (8.4) (see Fig. 8.5(b) and (d)). Overlaid occlusions occur between the relative foreground-background as defined in Section 3.4. These overlaid occlusions can be identified by searching for any DDPs in the previously identified occlusion layer, see Fig. 8.5(g). If any such DDPs exist then the occlusion mask for the overlaid occlusion can similarly be formed using (8.4). An example of an overlaid occlusion is shown in Fig. 8.5(h). All the occlusion layers in the LDI are formed in this manner.

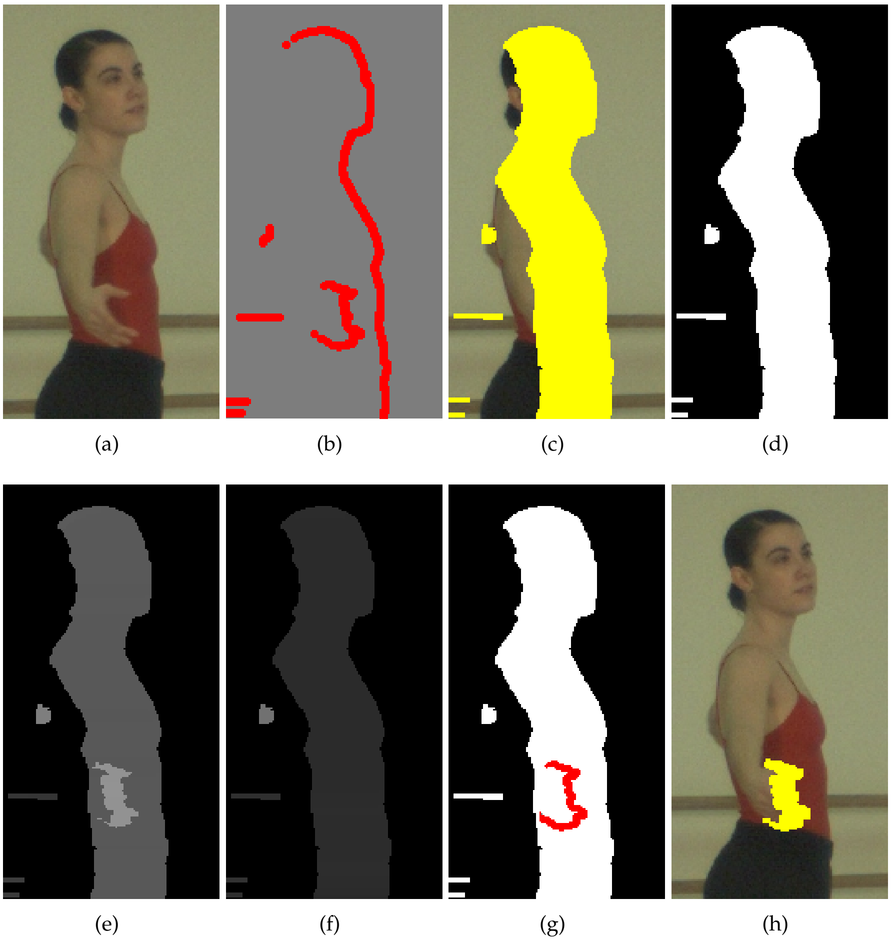


Figure 8.5: "Ballet" (frame28); (a) Original image; (b) Depth discontinuity pixels (in red color); (c) Occlusion in original image (in yellow color); (d) Occlusion mask; (e) Average depth threshold image; (f) Background depth threshold image; (g) Depth discontinuity pixels in former occlusion mask; (h) Overlaid occlusion.

Occlusion depth threshold

Identified occlusions belong to the background and should be filled with consistent background information. Hence depth threshold images are created from the DDPs along with the occlusion mask in order to facilitate occlusion inpainting with consistent background information. The threshold images contain the average and background depth values of the DDPs at the occlusion areas. The threshold images are named the average threshold image Z_A and the background threshold image Z_B respectively. These two images are defined by extrapolating the depth values of the DDPs into the occlusion areas. All pixel values within the occlusion area are set as functions of the DDP depth values at $\{\mathbf{f}_\mathrm{o}, \mathbf{p}_\mathrm{o}\}$:

$$\left.\begin{aligned} \mathrm{Z}_\mathrm{A}\left(\mathbf{q}\right) &= T_1, \\ \mathrm{Z}_\mathrm{B}\left(\mathbf{q}\right) &= \mathrm{Z}(\mathbf{p}_\mathrm{o}), \end{aligned}\right\} \quad \forall \mathbf{q} \in \overline{\mathbf{p}_\mathrm{o}\mathbf{r}_\mathrm{o}}, \tag{8.6}$$

Note that pixels on different lines $\overline{\mathbf{p}_\mathrm{o}\mathbf{r}_\mathrm{o}}$ are calculated with different DDPs. The pixels in the original view with depth values lthat are arger than the pixel values in the $\mathrm{Z}_\mathrm{A}(\mathbf{q})$ are belong to the foreground. Examples of Z_A and Z_B are shown in Fig. 8.5(e) and (f). The brightest region in Fig. 8.5(e) corresponds to the overlaid occlusion.

8.2.2 Hole-filling

The hole-filling method is here an extended version of the method introduced in Chapter 7. It uses depth constraints in the following steps: foreground-background boundary extraction, filling priority, and patch matching and filling indicated by red boxes in Fig. 8.6. The background data for computing the filling priority and patch matching use the proposed classification introduced in Section 8.2.1. A depth threshold for classification is derived from the depth discontinuity values, and so the proposed method ensures that the holes are filled from the background.

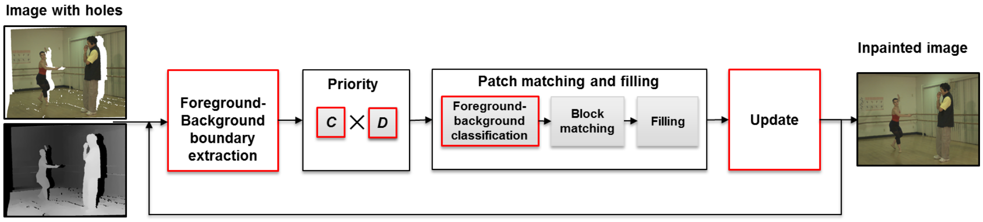


Image with holes
Foreground-Background boundary extraction
Priority
C
D
Patch matching and filling
Foreground-background classification
Block matching
Filling
Update
Inpainted image

Figure 8.6: Proposed depth-based inpainting method.

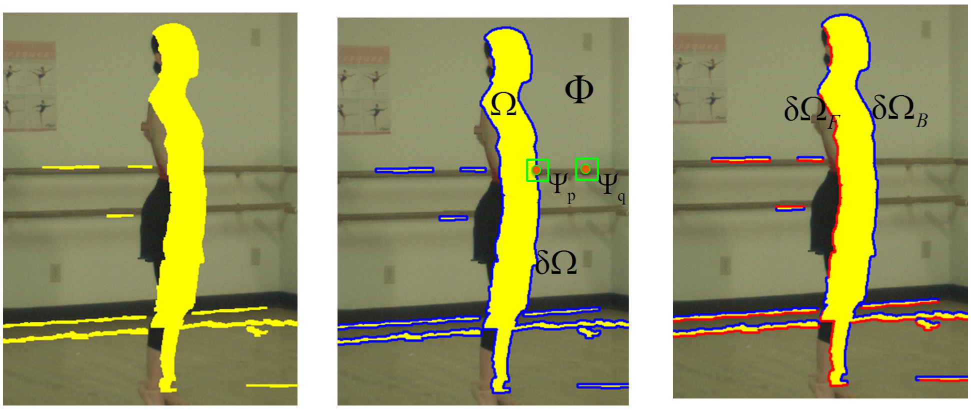
Foreground-Background boundary extraction

The hole boundary is classified into the foreground boundary $\delta\Omega_\mathrm{F}$ and the background boundary $\delta\Omega_\mathrm{B}$ using the depth threshold image $\mathbf{Z}_\mathrm{A}$ i.e., $\delta\Omega = \delta\Omega_\mathrm{F} \cup \delta\Omega_\mathrm{B}$, see Fig. 8.7 (c).

$$\delta\Omega_\mathrm{F} = \delta\Omega \;\; \forall \mathbf{p} : \mathbf{Z}\left(\mathbf{p}\right) > \max\left(\mathbf{Z}_\mathrm{A}\left(\mathbf{q}\right)|_{\mathbf{q}\in(\Psi_\mathbf{p}\cap\Omega)}\right), \tag{8.7}$$

$$\delta\Omega_\mathrm{B} = \delta\Omega \;\; \forall \mathbf{p} : \mathbf{Z}\left(\mathbf{p}\right) < \max\left(\mathbf{Z}_\mathrm{A}\left(\mathbf{q}\right)|_{\mathbf{q}\in(\Psi_\mathbf{p}\cap\Omega)}\right), \tag{8.8}$$

where Ω is hole region, $\Psi_\mathbf{p}$ is patch centered at a pixel $\mathbf{p}$.



(a) (b) (c)

Figure 8.7: Hole boundary extraction: (a) Texture image with holes; (b) Hole boundary with notations (boundary in blue color); (c) Depth guided boundary (foreground and background boundaries are in red and blue colors).

Priority

The filling priority is computed on the background boundary $\delta\Omega_{\mathrm{B}}$ is computed as the product of the confidence and data terms:

$$P(\mathbf{p}) = C_z(\mathbf{p}) \cdot D_k(\mathbf{p}), \tag{8.9}$$

where, $C_z(\mathbf{p})$ is the depth-based confidence term and $D_k(\mathbf{p})$ is the curvature data term at $\mathbf{p}$.

Depth-based confidence term: Confidence is one of the important terms that drives the filling inwards. Since the term computes the confidence irrespective of whether the pixels are in the foreground or background, it leads to inconsistent filling. To avoid this problem, unlike [AK13], the confidence term is only computed for the patches that contain the background information. The confidence term for the patch is computed using (7.5) in previous chapter, where the confidence for each pixel has been altered to be

$$C(\mathbf{q}) = \begin{cases} 0 & \text{if } (\mathbf{q} \in \Omega) \vee \left(\mathbf{Z}(\mathbf{q}) > \max\left(\mathbf{Z}_{\mathrm{A}}(\mathbf{r})\,|_{\mathbf{r}\in(\Psi_{\mathbf{p}}\cap\Omega)}\right)\right); \\ 1 & \text{otherwise.} \end{cases} \tag{8.10}$$

Curvature data term: A curvature data term is used similar to Section 7.2.2. Although the isophote and structure tensor data terms aim at propagating the linear structure into the missing regions, using such a data term implies that holes are filled with inconsistent structures due to the selection of strong gradients (see Fig. 8.8 (b) and (c)). To avoid selecting strongest structure selection, the data term contains the structure of a neighborhood by considering the curvature of the isophote $\kappa_{\mathbf{p}}$ through the pixel p:

$$D_{\kappa}(\mathbf{p}) = 1 - |k_{\mathbf{p}}|\,. \tag{8.11}$$

where $\kappa_{\mathbf{p}}$ is defined in (7.8). The analysis in Section 7.5.2 shows that the depth in the data term does not significantly influence the image quality, thus the data term is only calculated on the texture image.

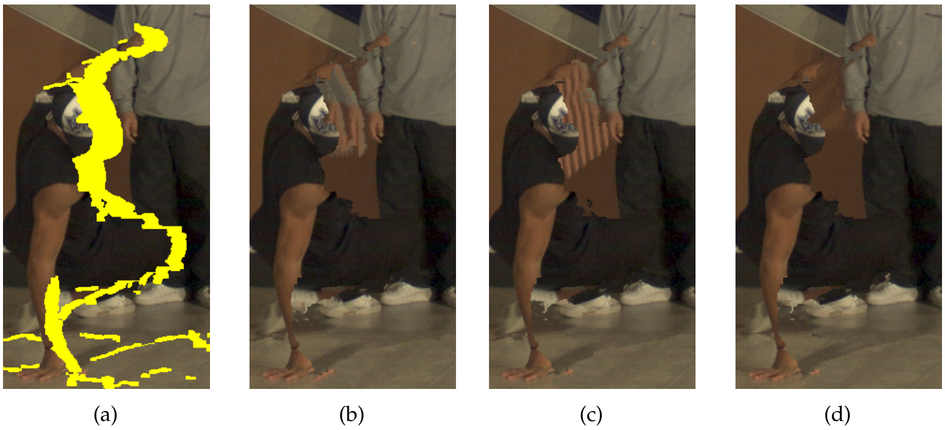


Figure 8.8: Consequence of different data terms: (a) Texture image showing holes; (b) Holes filled using Isophote data term [CPT04]; (c) Holes filled using structure tensor data term [AK13]; (d) Holes filled using proposed curvature data term.

Patch matching and filling

Foreground-background classification: It is experimentally shown in Section 7.5.2 that the source region classification is crucial for handling holes when the depth consists of several layers. Thus, the thresholds T_{U} and T_{L} are selected from the threshold images $\mathbf{Z}_{\mathrm{A}}$ and $\mathbf{Z}_{\mathrm{B}}$. The source patch must be searched for in the source region.

$$T_{\mathrm{U}} = \max\left(\mathbf{Z}_{\mathrm{A}}(\mathbf{q})\,|_{\mathbf{q}\in(\Psi_{\hat{\mathbf{p}}}\cap\Omega)}\right), \tag{8.12}$$

$$\Phi_{\mathrm{B}} = \Phi - \Phi_{\mathrm{F}}, \tag{8.13}$$

where Φ_{F} is the foreground source region, and where the depth values are larger than the depth threshold T_{U}. In the case of overlaid occlusions, the search is done in the background source region Φ_{B} selected between the depth thresholds T_{U} and T_{L} as overlaid occlusions that occur between two different depth regions.

$$T_{\mathrm{L}} = \min\left(\mathbf{Z}_{\mathrm{B}}\left(\mathbf{q}\right)|_{\mathbf{q}\in(\Psi_{\hat{\mathbf{p}}}\cap\Omega)}\right) - \chi, \tag{8.14}$$

where χ is a depth tolerance threshold that which allows to find a best match according to depth. In this process, the foreground pixels are removed from the target patch during block matching in order to avoid jaggedness, as in Section 7.5.3. Once the background source region is identified, source patches are searched using (7.11). Holes are then filled in both texture and depth using the method in Section 7.2.

Update

The proposed filling process is iterative, by which the target patch is filled with appropriate source patch data. In each iteration, the regions updated such that the filled target patch becomes part of the source region for the next iteration.

Unlike the methods presented in the literature, the proposed method fills each hole separately. This means that filling when neighboring holes exist in the vicinity of a patch size the method only fills the hole under consideration, not the neighboring hole. This is to allow the method fills the holes consistently with its neighboring data.

8.2.3 Warping LDI and merging

After the occlusion layers in the LDI are filled, they are warped to the virtual view position. The warped occlusion layers are then merged with the rest of the warped image from V+D. The merging uses Z-buffering to assure that the closer pixel is selected when two pixels are projected at the same location [TLLG09]. Note that the overlaid occlusion layers are warped using the background depth values stored in $\mathrm{Z_B}$ to ensure that the filled depth map does not create any holes within the inpainted region after projection.

8.2.4 Hole-filling post-processing

There may be holes in the warped image after the merging process. The holes are cracks and out-field-areas, Section 3.4. Consequently these holes are classified as cracks and large holes and to remove these artifacts in the virtual view, we apply the following steps in order to remove these artifacts in the virtual view:

1. Crack filling: Cracks are holes normally, which are smaller than the patch area. They are filled using simple background propagation. Such small holes have no priority during inpainting, since the target patch then contains foreground data. This implies that such patches with no-priority are filled after all of the remaining holes in the image are filled. Filling holes in this way would lead to inconsistent hole-filling, because it is less likely that a consistent texture can

be found fill the holes. Instead, these cracks are filled with simple background propagation.

2. Hole-filling Post-processing: The remaining holes mainly consist of out-of-field areas and are filled using the proposed inpainting method. No depth classification is required, since there is no information about the foreground at the borders of these holes, which makes depth classification redundant.

8.3 Methodology

Five MVD test sequences with various texture and depth characteristics were used to validate our proposed view synthesis method. The sequences are “Newspaper” from the 3DVC reference set, “Ballet”, “Breakdancers”, “Lovebird1”, and “Poznan Street”. The characteristics of the chosen sequences are detailed in Table 6.1.

Similar to the test setup in Chapter 7, objective metrics (YPSNR and MSSIM) and visual inspections were used to assess the quality of the proposed method.

The following parameter values were empirically selected for all test sequences: In the occlusion identification step, we used the occlusion threshold $\eta = 5$ pixels in Eq. (8.1) for small base lines, because holes that are smaller than a patch width affect the filling process in finding consistent textures. Moreover, selecting larger values of η would require an additional post-processing hole filling step after warping, as the holes smaller than the threshold are not located in the original view. In the patch matching step, a search region of 120x120 pixels and a patch size of 11x11 pixels were selected. In the case of overlaid occlusion filling, a patch size of 9x9 pixels is used based on the assumption that small areas can be effectively filled with a small patch size. The influence of search region and patch size is presented in [AK13]. The number of best patches were set $N_{\mathrm{b}} = 5$. The depth tolerance parameter χ in (8.14) controls the range of depth values in the block matching search. Larger values of χ allows holes to be filled with absolute background that can cause translucent problems. On the other hand, smaller χ values lead to inconsistent hole-filling, when the search region lacks sufficient depth values. Thus the depth tolerance is set to $\chi = 5$. The number of occlusion layers depends on occlusions in the scene; however, the number of layers inpainted by the proposed method is set to two.

There are several parameters involved in the classification and inpainting parts of the proposed method. We divide possible combinations into three experimental set-ups, each using a different set of key parameters for evaluation purposes:

A. Depth-based inpainting

B. Translucent disocclusions

C. Foreground-background classification

The following three subsections elaborate on the details of the above set-ups .

8.3.1 Depth-based inpainting

The evaluation of the depth-based inpainting method was performed both in 1D and general modes. A set of 100 frames were selected from the test sequences in order to show the consistency across the frames.

Results from the proposed method are compared to the results from the state-of-the-art methods M1-M5 from [vsr10, DS11, GMG11, AK13, WMG13]. Method M1 uses diffusion to fill holes in the warped image, whereas the remaining methods use the exemplar method to fill holes. Comparisons were performed for the available data from the given reference methods. Methods M2, M3 and M4 are used in the traditional scenario, i.e. when the hole-filling is applied in the virtual view. Method M5 is similar to our method where hole-filling applies in the original view. In this evaluation, M5 was used only in the general mode, since the method code for 1D mode is not available. Moreover, M5 only aims for the view consistency and does not provide a solution for filling out-of-field areas. Hence, in order to have a fair comparison, unfilled areas in the warped image in method M5 are filled using method M3, as M5 uses the inpainting method from M3. Method M3 does not produce results for all frames; since the results show color artifacts and unfilled holes in the inpainted images. Thus a few frames were excluded in the measurements to make the comparison fair. The discarded frames are from "Ballet" view 6, "Breakdancers" view 6 and "Poznan Street" view 3.

8.3.2 Translucent disocclusions

The occurrence of translucent disocclusions depends on the scene, and therefore on the depth map characteristics. Thus we have selected a subset of the "Ballet" sequence, which suffers from translucent disocclusions. The chosen subset includes five frames from the camera view position 5, rendered to view 4 and view 7 positions respectively.

In this comparison we only used the proposed method, since the translucent disocclusion is a variable. As the overlaid occlusion in the original view corresponds to disocclusion in the virtual view, the comparison was performed by rendering, both with and without overlaid occlusion filling. In the result figures in the next section, the method with overlaid occlusion filling is labeled as Translucent Disocclusion Handling (TDH), whereas the other is labeled No Translucent Disocclusion Handling (NTDH).

8.3.3 Foreground-background classification

A subset of the "Ballet" sequence that possesses many depth layers was selected, in order to test the influence of the foreground-background classification in the patch matching process. The chosen subset includes five frames from the "Ballet" sequence view 5, rendered to view 3 and view 4 positions respectively, which have objects at multiple depths. Such multiple depths constitute relative foreground-background occlusions and may therefore also produce translucent disocclusions, which was investigated in Section 8.3.2. The evaluation of the foreground-background evaluation used the same sequence as in Section 8.3.2 but different frames.

In this evaluation, the foreground-background classification is a variable; so that the remaining parameters in the inpainting process are kept unchanged. This means that the foreground-background classification part in the proposed method is exchanged with the classification procedure from reference methods. Two reference depth classification methods were used in the evaluation. They are from method M4 and from Choi et al.(M6) [CHS13]. Methods M1-M3 have not used the depth classification in the inpainting process and the code for the classification is not separately available in the case of method M5. Therefore, depth classification evaluations are limited to the methods M4 and M6.

8.4 Results and analysis

8.4.1 Depth-based inpainting

The objective evaluation results of the depth-based inpainting quality are presented in Tables 8.1 and 8.2 respectively. In addition to the average YPSNR and MSSIM, the objective measurements of 100 frames are shown in Fig. 8.9. The results in Table 8.1 and 8.2 demonstrate that the proposed method performs better than the state-of-the-art reference methods. In 1D mode, the proposed method performed better than depth-based inpainting methods M2 and M4, but the objective quality was slightly reduced compared to the M1 and M3 methods. Two reasons can explain the results from the depth-based inpainting methods comparison:

i. The depth-based inpainting method M3 uses the true depth during the inpainting process, whereas the proposed method operates on general settings so that the warped depth is estimated along with the texture. Despite that, holes were filled with an inconsistent texture in our proposed method due to insufficient background information in the neighborhood. This problem usually occurs when the depth map quality is poor. Especially the "Newspaper" sequence has that characteristic.

ii. A major portion of the holes in the sequences "Newspaper" and "Poznan Street" are out-of-field-areas. As our approach does not apply the depth classification in filling out-of-field-areas, inconsistent filling is possible when foreground objects are present at the borders.

The second reasoning is valid for method M1. Although the holes are not filled with consistent textures using M1, it shows slightly better objective results in the1D case because the holes are small and appear between similar textures. This result implies that the diffusion can be a valid choice for filling small holes and homogenous regions. It is worth noting that objective measurements for ill posed problems such

Table 8.1: Objective Quality Evaluation: Averaged YPSNR.

Test sequence	Proposed	M5	M4	M3	M2	M1
Ballet v5→v4	**31.95**	28.75	31.08	28.43	28.68	27.34
Ballet v5→v6	**29.01**	26.80	27.71	27.63	26.52	26.37
Breakdancers v5→v4	**30.97**	29.96	29.28	30.03	30.46	30.76
Breakdancers v5→v6	**31.38**	30.90	30.98	31.08	30.54	31.26
Poznan street v5→v3	28.07	-	27.77	-	27.67	**28.20**
Lovebird1 v6→v4	**25.14**	-	25.04	25.13	24.94	24.94
Newspaper v4→v6	23.24	-	20.55	**23.60**	23.13	23.32

as the inpainting quality are still a challenging problem with no existing metric that reflects the visual quality. Thus, the results are also presented for visual comparison, see Fig. 8.10 to 8.15 .

The results in Fig. 8.10(f) show how the proposed method performs a proper propagation of the structure into the holes and ensures spatial consistency with neighboring textures. The results in Fig. 8.11 further demonstrate the reconstruction of the consistent background texture by the proposed method,even when the data are missing between the two foreground objects (see missing data between the small poles in Fig. 8.11(e)).

In the general mode, the proposed method demonstrate a clear performance improvement over the reference methods, especially at the translucent disocclusions areas in the "Ballet" sequence (see Fig. 8.12(g)). Although M5 has slightly filled the translucent disocclusions, uncovered parts are still left unfilled (see Fig. 8.12(f)). Moreover, when the disocclusions occur between the relative foreground-background, the reference methods fail to reconstruct the missing data while the proposed method ensures that there are no disocclusions ((see Fig. 8.13(g)) at the man's hand and at the woman's legs). The results of the proposed method in Fig. 8.14(g) and Fig. 8.15(g) show the consistent filling of the background structures when there are strong gradients around holes. In the same scenario, the reference methods fill background structures with the strongest gradients, which looks unnatural and inconsistent with the neighboring background (see Fig. 8.14(c) to (f) and Fig. 8.15(c) to (f)).

8.4.2 Translucent disocclusions

Table 8.3 presents an objective evaluation of translucent disocclusions. YPSNR and MSSIM measurements consistently demonstrate that the objective quality is improved

Table 8.2: Objective Quality Evaluation: Averaged MSSIM.

Test sequence	**Proposed**	**M5**	**M4**	**M3**	**M2**	**M1**
Ballet v5→v4	**0.8724**	0.8631	0.8662	0.8620	0.8504	0.8605
Ballet v5→v6	**0.8495**	0.8439	0.8395	0.8335	0.8274	0.8493
Breakdancers v5→v4	**0.8255**	0.8244	0.8250	0.8188	0.8182	0.8213
Breakdancers v5→v6	0.8214	0.8220	0.8184	0.8211	0.8154	**0.8238**
Poznan street v5→v3	0.8341	-	0.8287	-	0.8279	**0.8393**
Lovebird1 v6→v4	0.8606	-	0.8572	0.8595	0.8551	**0.8655**
Newspaper v4→v6	0.8462	-	0.8330	0.8426	0.8367	**0.8525**

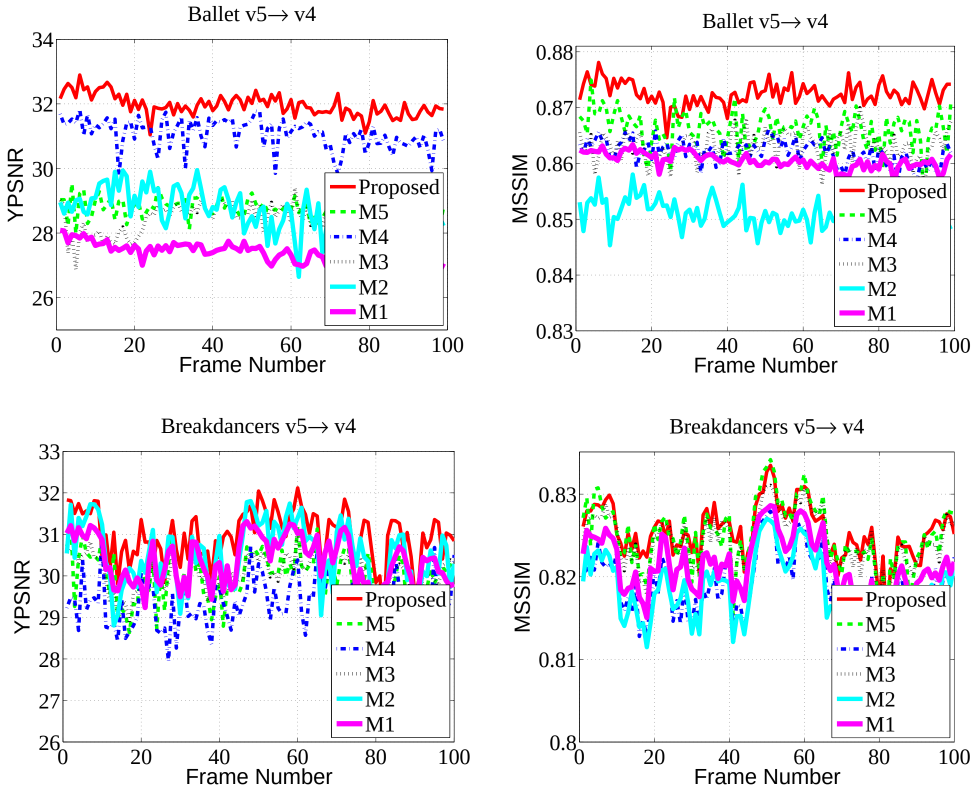
by the translucent disocclusion handling. The improvements are still in the decimal range since the disocclusions occupy only a very small portion of the image. Rendered results in Fig. 8.16(b) and (c) show the difference in the case of translucent disocclusion handling (TDH) and no translucent disocclusion handling (NTDH). Results look unnatural with NTDH because the background is seeping through the foreground, see Fig. 8.16(b), (e) and (h). Results highlight the importance of translucent disocclusion handling for improving the rendered image quality. As the occurrence of this problem depends on the scene and the depth map characteristics, any sequence that would have these properties will benefit by identifying and inpainting them.

Table 8.3: Translucent disocclusion objective quality evaluation

Ballet Seq	**Averaged YPSNR**		**Averaged MSSIM**	
	TDH	NTDH	TDH	NTDH
v5→v4	**32.13**	32.05	**0.8727**	0.8724
v5→v7	**27.84**	27.44	**0.8172**	0.8158

8.4.3 Foreground-background classification

Results for the objective quality assessment of the foreground-background classification are presented in Table 8.4. The objective measurements YPSNR and MSSIM consistently demonstrate that the proposed foreground-background classification im-



(a) (b)

Figure 8.9: View synthesis with depth-based inpainting objective quality evaluation: (a) YPSNR; (b) MSSIM.

proves the objective quality. Further, the rendered results are shown in Fig. 8.17 for visual comparison. When filling with the reference methods, the structure is disconnected and inconsistent with the neighboring background (see Fig. 8.17(b) and (c) at the woman's right leg and the right side of the man at the wooden bar). Fig. 8.17(d) illustrates how the proposed classification performs better than the reference classification methods. Understandably, a spatially consistent foreground-background is required to make the disocclusions filling plausible.

8.5 Concluding remarks

An alternative formulation of the view rendering method using layered depth image (LDI) and inpainting has been presented in order to improve spatial consistency and reduce rendering artifacts. Moreover, the method can offer view consistency by

Table 8.4: Objective quality evaluation using different depth classification algorithms

Ballet Seq	**Averaged YPSNR**			**Averaged MSSIM**		
	Proposed	M4	M6	Proposed	M4	M6
v5→v4	**32.09**	31.59	31.58	**0.8731**	0.8719	0.8689
v5→v3	**27.99**	27.46	27.35	**0.8402**	0.8375	0.8308

inpainting occlusions in the farthest view and use that information for closer views. The proposed depth-based LDI inpainting method ensures that no disocclusions appear in the virtual view. Furthermore, the proposed foreground-background classification ensures filling of occlusions consistent with the background. The proposed method computes the filling priority solely on the background, in order to propagate consistent structure details with respect to its neighboring background. The objective test results and visual inspection consistently demonstrate that the proposed method produces high quality virtual views, especially well coherent at the translucent disocclusions for the tested sequences. The results demonstrate the importance of translucent disocclusion handling and foreground-background classification in improving the virtual view image quality.

8.5.1 Contributions

The author's contribution to this chapter was both introducing an alternative rendering method using LDI and inpainting, as well as the evaluation of key parts in the rendering. The results of this chapter has been summarized in the manuscript Paper VI.

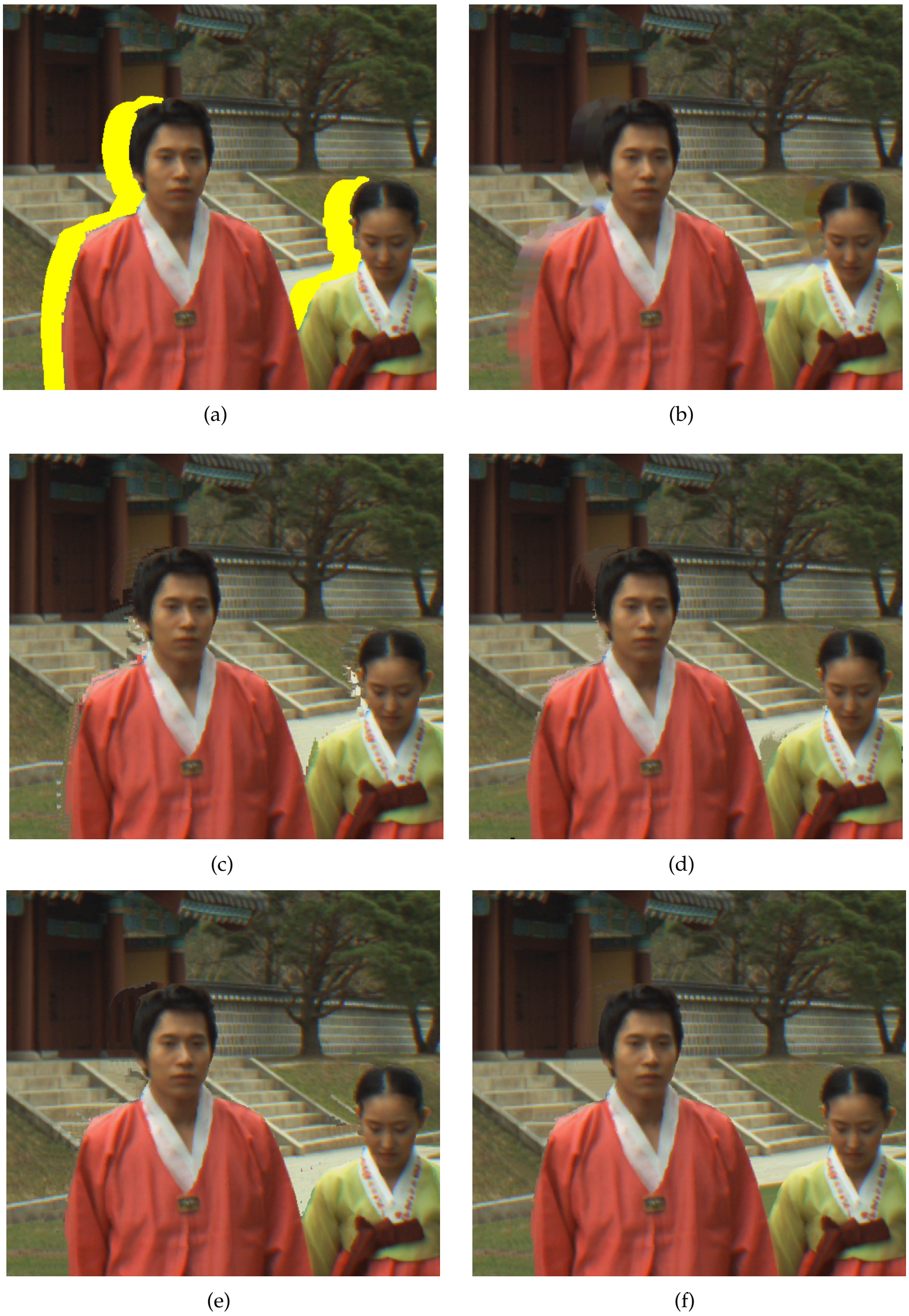
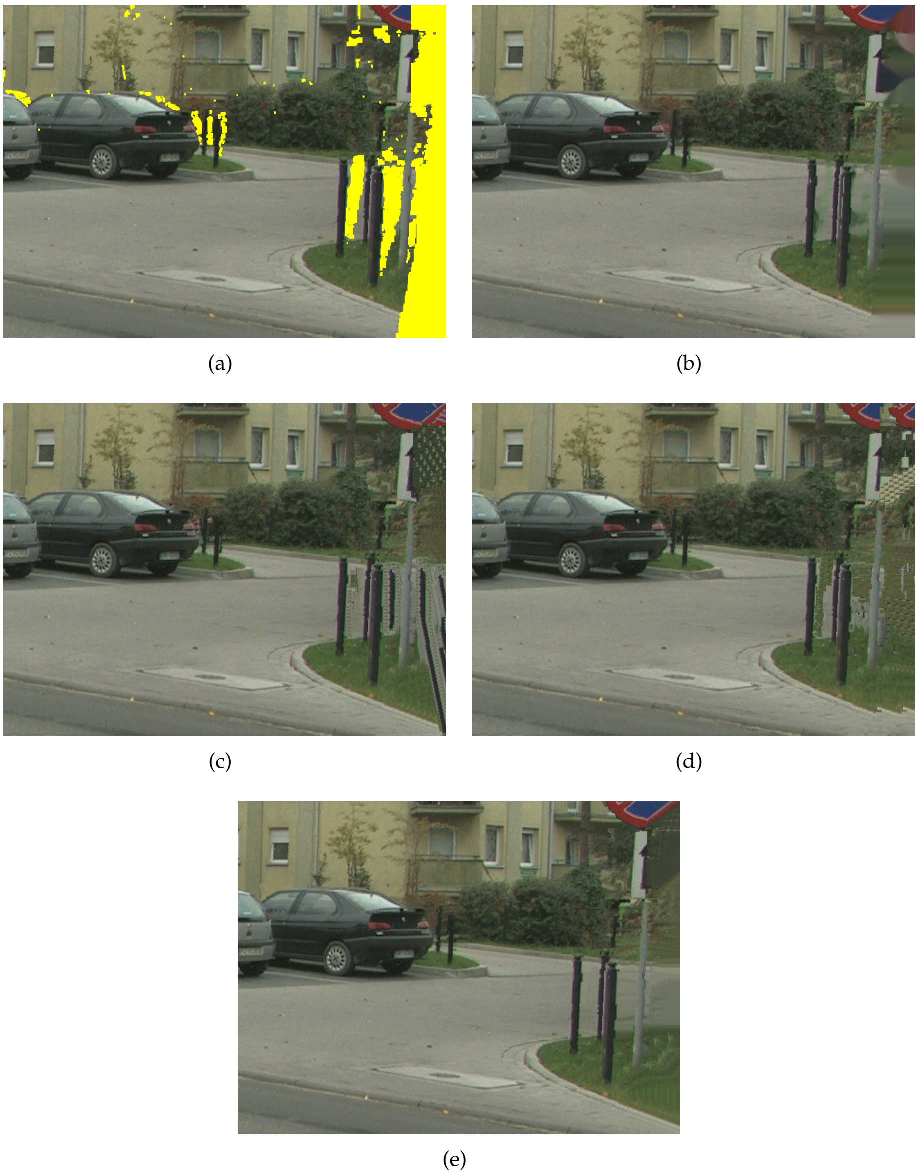


Figure 8.10: Depth-based inpainting of“Lovebird1” v6→v4 frame192: (a) Warped image with holes marked in yellow color; (b) M1 [vsr10] ; (c) M2 [DS11] ; (d) M3 [GMG11] ; (e) M4 [AK13] ; (f) Proposed method.


(a)

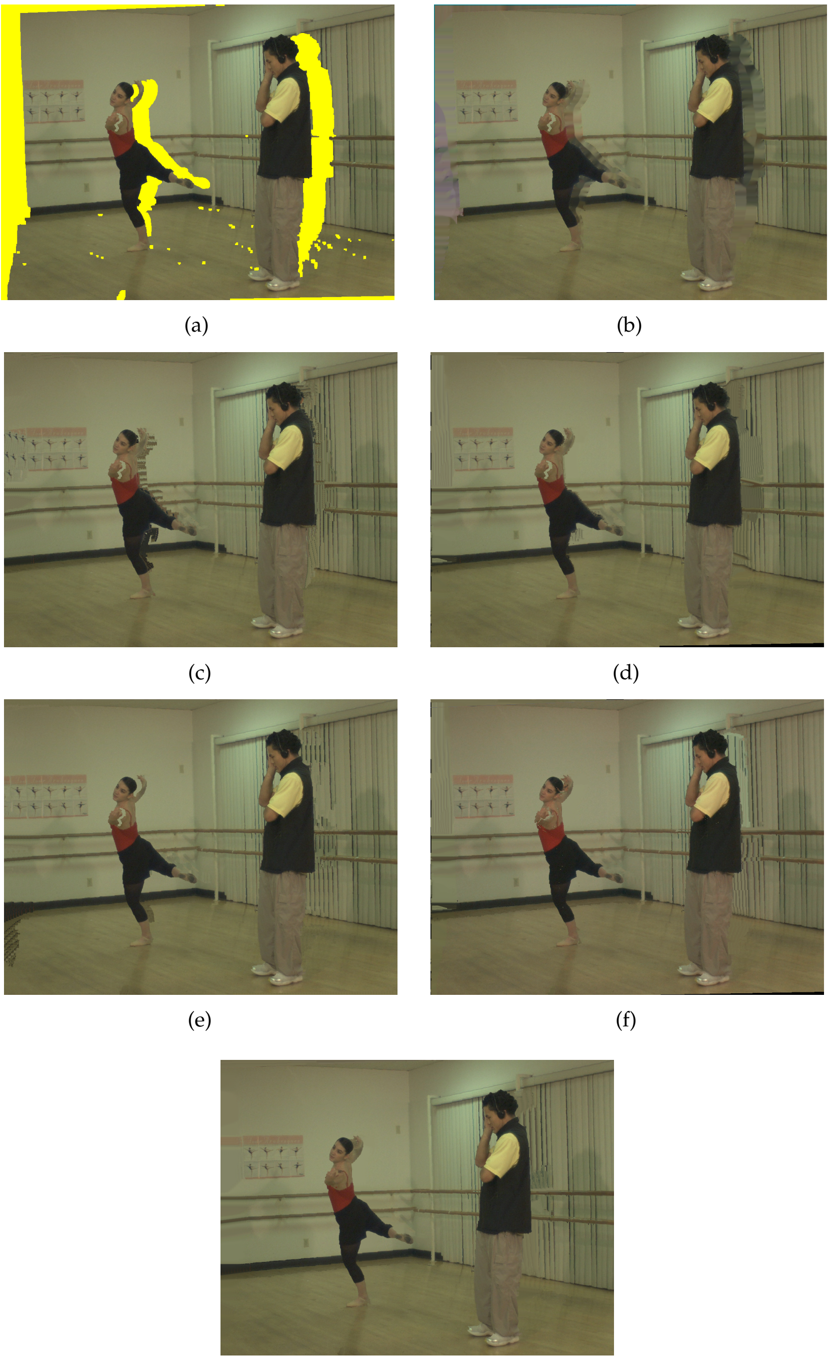
(b)

(c)

(d)

(e)

Figure 8.11: Depth-based inpainting of “Poznan street” v5→v3 frame192: (a) Warped image with holes marked in yellow color; (b) M1 [vsr10]; (c) M2 [DS11]; (d) M4 [AK13]; (e) Proposed method.



(a)

(b)

(c)

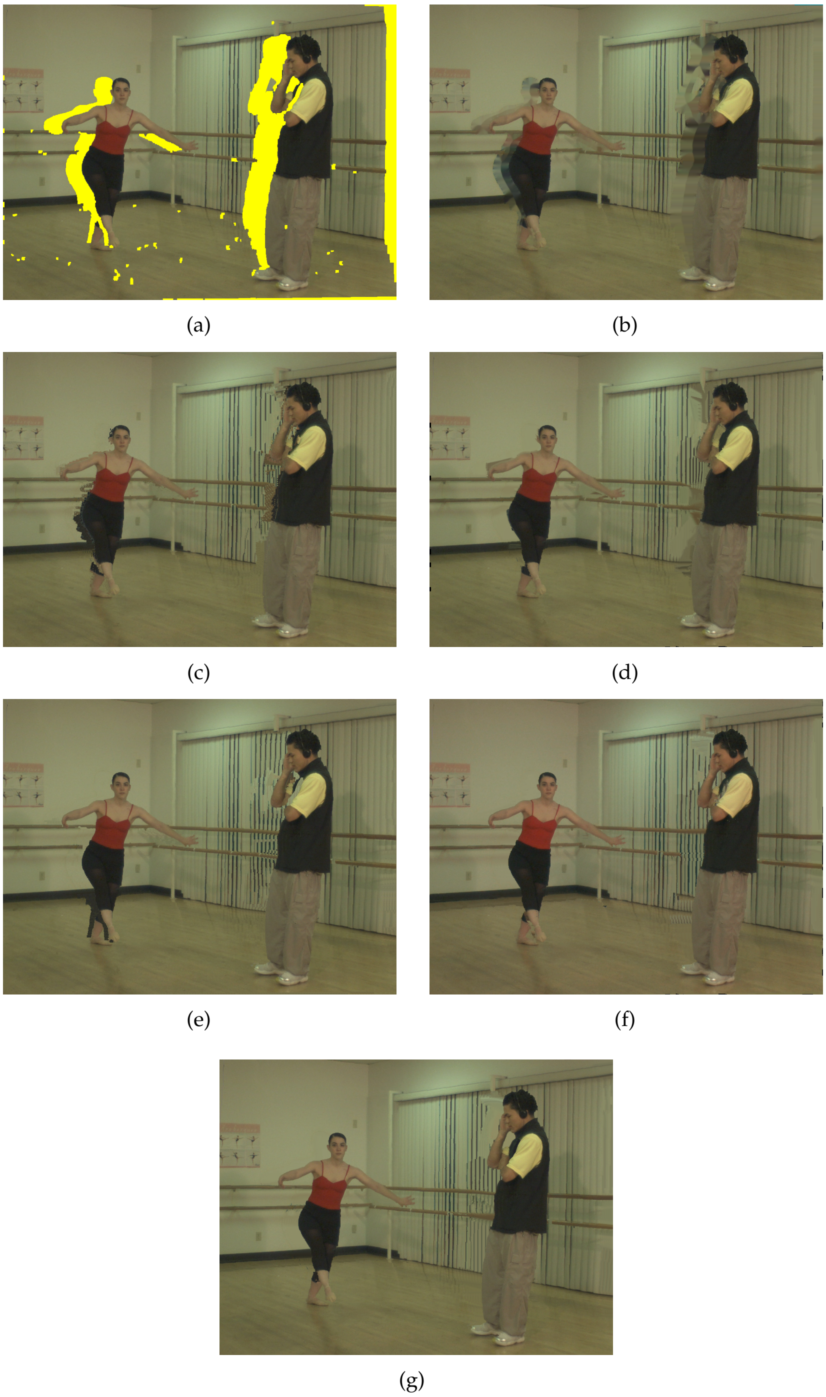
(d)

(e)

(f)

(g)

Figure 8.12: Depth-based inpainting of "Ballet" v5→v4 frame19: (a) Texture image with holes marked in yellow color; (b) M1 [vsr10]; (c) M2 [DS11]; (d) M3 [GMG11]; (e) M4 [AK13]; (f) M5 [WMG13]; (g) Proposed method;



(a) (b) (c) (d) (e) (f) (g)

Figure 8.13: Depth-based inpainting of “Ballet” v5→v6 frame19: (a) Texture image with holes marked in yellow color; (b) M1 [vsr10]; (c) M2 [DS11]; (d) M3 [GMG11]; (e) M4 [AK13]; (f) M5 [WMG13]; (g) Proposed method;

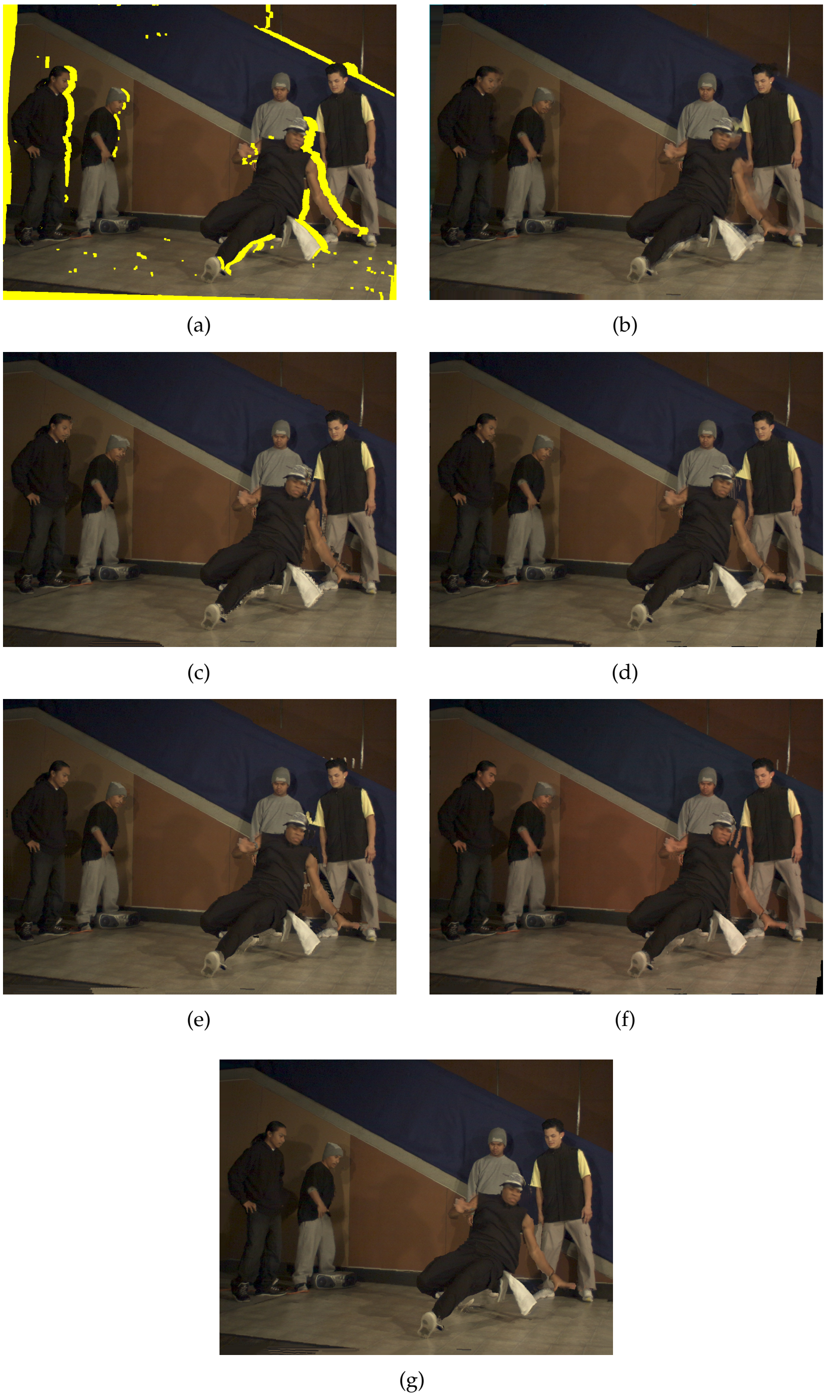
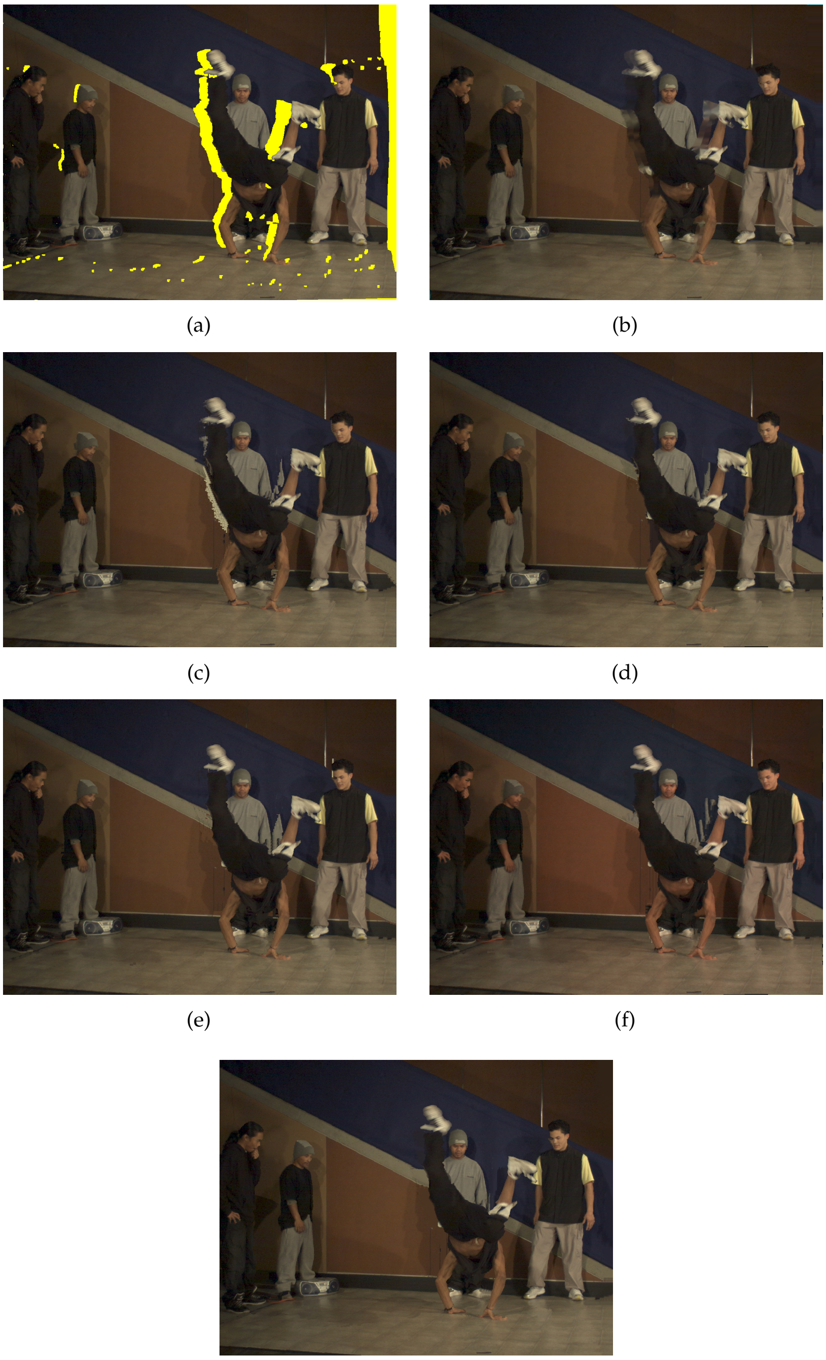


Figure 8.14: Depth-based inpainting of “Breakdancers” v5→v4 frame31: (a) Texture image with holes marked in yellow color; (b) M1 [vsr10]; (c) M2 [DS11]; (d) M3 [GMG11]; (e) M4 [AK13]; (f) M5 [WMG13]; (g) Proposed method;



(a)

(b)

(c)

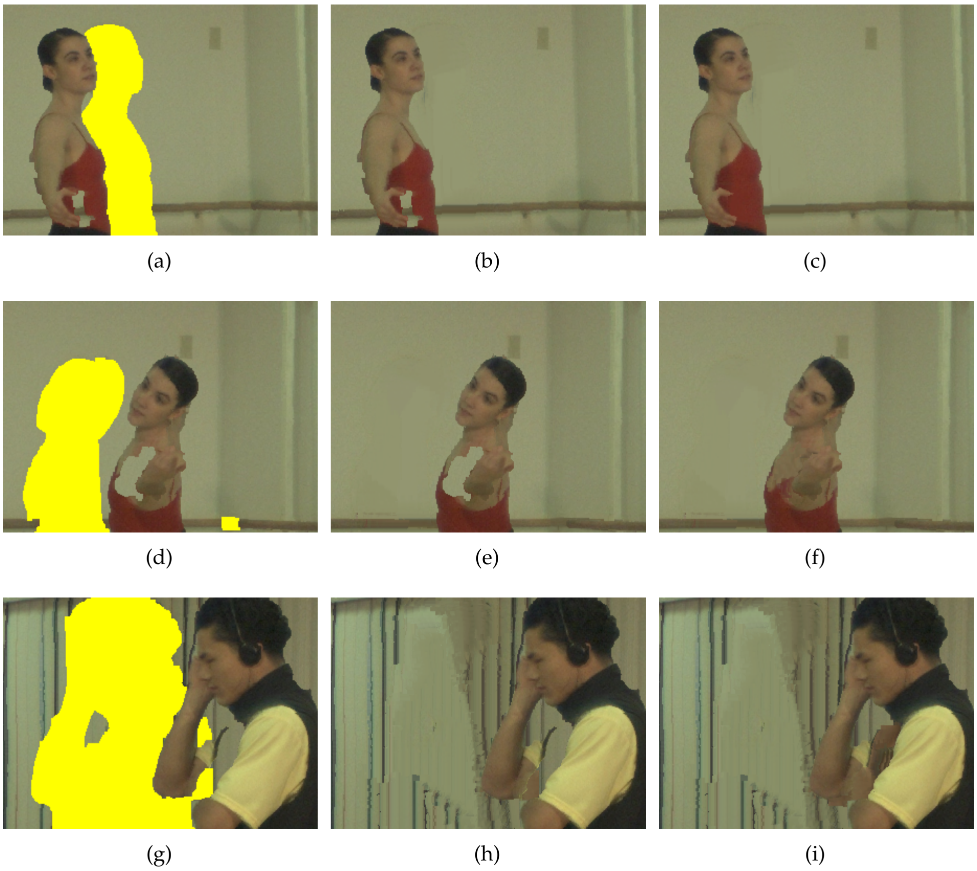
(d)

(e)

(f)

(g)

Figure 8.15: Depth-based inpainting of “Breakdancers” v5→v6 frame75: (a) Texture image with holes marked in yellow color; (b) M1 [vsr10]; (c) M2 [DS11]; (d) M3 [GMG11]; (e) M4 [AK13]; (f) M5 [WMG13]; (g) Proposed method.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 8.16: Translucent disocclusion: (a) Warped image “Ballet” v5→v4 frame28 holes marked in yellow color; (b) No translucent disocclusion handling (NTDH); (c) Translucent disocclusion handling (TDH); (d) Warped image “Ballet” v5→v3 frame37; (e) NTDH; (f) TDH; (g) Warped image “Ballet” v5→v3 frame37 ; (h) NTDH; (i) TDH.

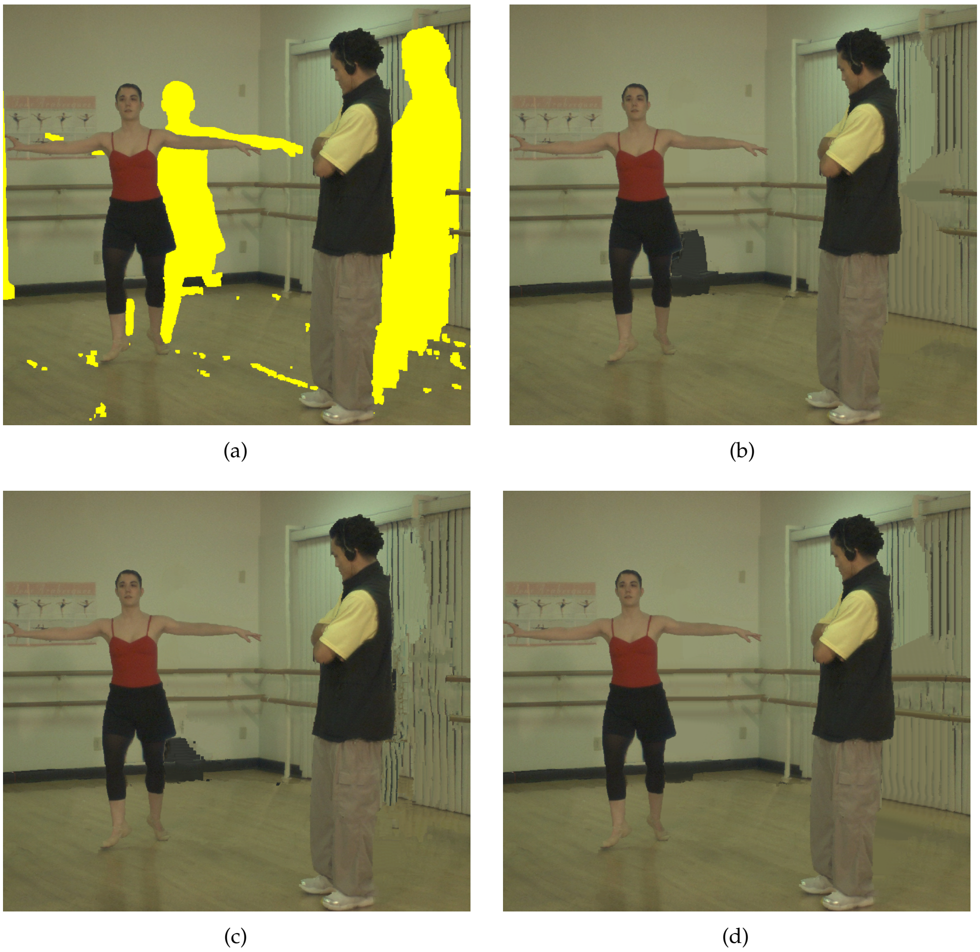


Figure 8.17: Depth classification: (a) Warped image "Ballet" v5→v3 frame37 holes marked in yellow color; (b) Classification from M4; (c) Classification from M6; (d) Proposed method.

Chapter 9

Conclusions and Future Work

This final chapter provides conclusions of the dissertation with regards to the view rendering topic. An overview of the content is presented and the outcome of the presented research is discussed together with the impact of the work. Suggestions for future work are then offered with possible extensions and applications of the current approaches. Finally, the chapter is conclude with a short discussion on the ethical considerations of the work.

9.1 Overview

The purpose of this dissertation is to improve the 3D video quality experience. In this context, high quality view rendering methods are essential. This dissertation has fulfilled its purpose by investigating, proposing, and evaluating novel rendering methods for 3D video formats video-plus-depth (V+D) and multiview-video-plus-depth (MVD). A new edge-aided rendering method for V+D and MVD was proposed. This method introduces the edge-pixels that enable an interpolation of the background colors into disoccluded regions. The experimental results demonstrated that the edge-aided rendering method performs slightly better than state-of-the-art methods in objective comparison and showed similar quality in subjective assessments. Moreover, the edge-aided rendering method offers a straightforward approach to handle depth-image-based rendering (DIBR) artifacts appearing in virtual views, whereas other state-of-the-art methods apply specific tools to minimize the rendering artifacts. The details about the edge-aided rendering method were provided in Chapter 6.

Because proper handling of disocclusions is a significant challenge in DIBR methods, a new depth-based inpainting method has been proposed to solve the disocclusion problems in the virtual view; this method is described in Chapter 7. The proposed method effectively uses the warped depth and texture information to fill the disocclusions. It computes the filling priority using only background information and fills the disocclusions using necessary data from the background. Experimental results have confirmed that the proposed depth-based inpainting method performs better than state-of-the-art inpainting methods.

A new rendering method using layered depth image (LDI) and inpainting was proposed by using the findings from Chapters 6 and 7. Details were described in Chapter 8. The proposed method avoids the disocclusion problems by inpainting the corresponding occlusions in the original view. By this procedure no disocclusions appear but views with spatial consistency are achieved. The results from the experiments show that the proposed rendering method using LDI and inpainting improves the spatial consistency, especially for translucent disocclusions and offers view consistency.

The use of the proposed methods satisfies the purpose of the dissertation by providing high quality rendered virtual views in order to improve 3DV quality.

9.2 Outcome

In Section 1.4 six objectives were defined in order to attain the aim of this dissertation. Below is a concise summary of how these objectives were addressed and the outcome with regards to each of the objectives:

O1 *To investigate rendering artifacts in the DIBR method.*

The problems associated with the DIBR method were examined. Typical artifacts associated with the DIBR method were identified as: ghosting artifacts, holes (cracks, disocclusions, out-of-field areas) and translucent artifacts (cracks, disocclusions). Furthermore, reference methods to address the DIBR problems and their consequences were investigated showing that the reference methods use several tools to address the artifacts. This study on the DIBR artifacts and state-of-the-art DIBR reference solutions were presented in Chapter 3 and Chapter 5. How these reference solutions and the methods proposed in this thesis reduce the identified DIBR artifacts is presented in Chapter 6.

O2 *To propose an alternative rendering solution that reduces artifacts.*

Based on the findings from Objective O1, an edge-aided rendering method was proposed to address the rendering artifacts. This method allows producing extrapolated and interpolated views from the V+D and MVD formats. This method utilizes so called edge-pixels, projected locations and interpolation to produce virtual views. Specifically, the introduced edge-pixels allow interpolating background colors into holes. Furthermore, the method merges views warped from different V+D into a single rendered view. Importantly, the proposed method fully avoids the post processing that is required to address each artifact separately in other methods. Objective assessments confirmed an improved image quality of each view using the proposed straight forward method. The proposed edge-aided rendering method for V+D and MVD formats was presented in Paper I and II, and included in Chapter 6 of this dissertation.

O3 *To investigate and analyze perceived visual quality of the rendered virtual views.*

In addition to the objective assessments of the virtual views, a subjective experiment was performed to investigate the visual quality of the rendered virtual views. In this experiment, the standard test procedure pair comparison (PC) from ITU-T Rec. P.910 was selected. The test was performed with 16 naive subjects by presenting images rendered using different methods and asking the subjects for their preferences. The results from the test demonstrate that the edge-aided rendering method achieves similar subjective quality as state-of-the art methods. This at a lower conceptual complexity, as mentioned in Objective O2. The results were presented in Paper II, and in Chapter 6 of this dissertation.

O4 *To investigate hole-filling methods in the rendering process and to propose a depth-based inpainting method to reduce artifacts.*

Different types of hole-filling methods were examined in the context of disocclusion handling: interpolation methods, diffusion-based inpainting methods and exemplar-based inpainting methods. Furthermore, the limitations and problems associated with existing hole-filling methods were investigated. The reference methods still suffer from foreground propagation and inconsistent background filling. The respective Investigations were presented in Chapter 5 of this dissertation.

Based on these findings, a depth-based curvature inpainting method was proposed in Paper III to address the disocclusion problems in the virtual views.

The method utilizes the warped depth information, a curvature data term and guides the inpainting process to propagate consistent structure according to the appropriate depth level. Objective metrics and visual assessments showed that the proposed method favors background propagation during the hole-filling process compared to the reference methods, as presented in Chapter 7.

O5 *To investigate the influence of depth information at various steps of the inpainting process and to propose a solution that address depth-based inpainting artifacts.*

Three key steps were identified in the inpainting process: boundary extraction, data term computation and patch matching. The influence of depth at each step was investigated. Depth-based inpainting artifacts were examined and categorized into background-leaking and jaggedness. Using the investigation results, an improved hole-filling method was proposed that relies on a depth edge-based source region classification. Objective metrics and visual assessments showed that the proposed method eliminated the categorized inpainting artifacts. The influence of depth in different steps of inpainting was presented in Paper IV. The depth edge-based source region classification method was proposed in Paper V, and is presented in Chapter 7 of this dissertation.

O6 *To investigate a spatial and view consistent rendering solution that avoids disocclusions in the rendered view.*

Based on the investigated impact of depth on the inpainting process from Objective O5, an alternative solution was proposed in Chapter 8 to address the disocclusions in the virtual views using LDI and inpainting. The proposed rendering method using LDI and inpainting avoids the disocclusion problems by filling corresponding occlusions in the original view. Furthermore, the influence of foreground-background classification in the inpainting and translucent disocclusion handling in the virtual view quality was examined. The results confirmed that the proposed method improves spatial consistency, especially at translucent disocclusions. Moreover, the method can offer view consistency by inpainting occlusions that would cause disocclusions in the farthest view and use that information for closer views. The results of Chapter 8 have been summarized in Manuscript VI.

9.3 Impact

As described in Section 1.1, high quality virtual views are required to provide better 3D video quality from a limited number of input views. Three solutions were proposed in this dissertation to improve the virtual view quality. The first was an edge-aided rendering method that utilizes low complex artifact handling, which makes it more suitable for 3D video mobile applications where a small rendered view base line is used [GAC$^+$11]. The other two target larger base lines and use inpainting to effectively replicate the missing data in the virtual view. These solutions can provide convincing high quality virtual views for 3DTV and FTV applications provided by home-entertainment setup box systems, where more processing power is available for the required calculations.

More explicit knowledge about the scene depth and the DIBR process and its artifacts is fundamental to improve any inpainting method. For this an artifact classification was proposed that identifies a set of different artifacts and their cause. Combining the identified classes into different groups allows for either low-complex processing that handles similar artifacts in a general way, or more elaborate solutions that process each artifact type in a specific way. Other class combinations than those presented in this dissertation may be suitable for applications where artifacts could led to severe consequences, e.g. in medical imaging, surveillance, and remote control of unmanned vehicles.

9.4 Future work

The following is a list of possible future work:

9.4.1 Rendering enhancements

When calculating the filling priority the data term differs from the confidence term in that it is related to the structure properties around the center pixel and not the complete patch. A curvature data term based on computed gradients using neighboring pixel differences may not efficiently extract the overall structure in the complete patch. This may be handled by employing a structure tensor with a curvature that would reflect structure details in the entire neighborhood of the center pixel.

The proposed approach demonstrated that a foreground-background classification plays a key role in producing high quality virtual views. Any classification method that gives more accurate depth labeling will directly enable better inpainting results with higher quality rendered views. In addition, performing the foreground-background classification only once, before LDI generation, would simplify the rendering method.

9.4.2 Temporal extensions

The result of hole-filling, either by interpolation or inpainting, is highly affected by the pixels surrounding the hole. Subsequent images in a video sequence may have similar holes, yet different pixel surroundings. The content of a filled hole might change from one image to another causing annoying flickering artifacts. So considering temporal properties of the video sequence is important to produce time consistent filling of all holes. A possible way to achieve this is by effectively using the available temporal information and extending previous inpainted information, instead of inpainting every image separately.

A possible approach possible approach to improve spatio-temporal consistency of rendered virtual views could be to build a static scene sprite of occlusion data, classify holes as either static or dynamic, and inpaint static holes once but dynamic holes continuously. Also including a buffer would allow dynamic holes to be filled with true data as may be revealed in future images, whereas static holes would retain their filled structure over time, thus reducing flickering.

9.4.3 Quality evaluation methodology

3D Video (3DV) quality assessment with regards to view rendering is still being researched [BPC$^+$11]. An overview of the recent developments and methods used to evaluate the rendered view from DIBR has been presented in [BCMP12, BBC$^+$15]. However, 3DV quality metrics have not yet fully incorporated depth-based inpainting problems like structure discontinuity, background-leaking and jaggedness. Hole-filling is an ill-posed problem that does not have a unique solution. Full-reference metrics that have access to the original image for comparison are not directly applicable for the inpainting case. When inpainting is performed there is no ideal disocclusion filling to strive for; as opposed to the case of image compression or processing, these data were never available. Instead of trying to replicate ground truth data it is more relevant to achieve spatial-, depth-, view-, and temporal consistencies. Hence, quality evaluation metric targeting inpainting results should possibly include other aspects than what traditional full-reference metrics do.

An important aspect to consider is that conventionally, image quality evaluation metrics have been applied to the whole image, so also recently proposed metrics for DIBR artifacts. In the case of inpainting the number of pixels affected by the process is generally significantly smaller than the total number of pixels. Metrics that normalize the result with respect to image size then risk to not be able to clearly identify distortion caused by the inpainting alone. Normalizing with respect to the number of pixels that are representing filled in holes address this, as the inpainting results are then the only things being measured. However, how to make such a metric highly correlated with perceived quality may be a challenging research topic.

9.5 Ethical considerations

As it is the researcher's ethical responsibility to be aware of the implications of the research, ethical issues have also been considered while conducting this research. The research presented in this work is to improve 3D video quality in order to create an immersive user experience. Special care was taken while conducting visual quality tests, where participants were allowed to stop the test if they felt any discomfort. Moreover, the test subjects voluntarily participated in the subjective test, which was conducted according to standard procedures [ITU08, ITU12].

Inpainting methods produce texture for missing regions in the images from the available data, meaning that the synthesized data may not be matched to the true data, except specific cases. So hole-filling methods may not be applicable for non-entertainment applications such as medical applications, where the filled information is somehow used for further processing or analysis, due to the risk of drawing erroneous conclusions about the visually plausible data.

Bibliography

[AFCS11] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro. A variational framework for exemplar-based image inpainting. *International Journal of Computer Vision*, 93(3):319–347, July 2011.

[AK13] I. Ahn and C. Kim. A novel depth-based virtual view synthesis method for free viewpoint video. *IEEE Transactions on Broadcasting*, 59(4):614–626, December 2013.

[BBC$^+$15] F. Battisti, E. Bosc, M. Carli, P. Le Callet, and S. Perugia. Objective image quality assessment of 3d synthesized views. *Signal Processing: Image Communication*, 30(0):78 – 88, 2015.

[BBG08] B. Barenbrug, R.-P. M. Berretty, and R. K. Gunnewiek. Robust image, depth, and occlusion generation from uncalibrated stereo. In *Society of Photo-Optical Instrumentation Engineers (SPIE) 6803*, pages 68031J–68031J–8, 2008.

[BBS01] M. Bertalmio, A. L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceeding IEEE Computer Vision and Pattern Recognition (CVPR*, pages 355–362, 2001.

[BCMP12] E. Bosc, P. Le Callet, L. Morin, and M. Pressigout. Visual quality assessment of synthesized views in the context of 3D-TV. In *3D-TV System with Depth-Image-Based Rendering Architectures, Techniques and Challenges*, pages 439–474. Springer, 2012.

[BPC$^+$11] E. Bosc, R. Pepion, P. Le Callet, M. Koppel, P. Ndjiki-Nya, M. Pressigout, and L. Morin. Towards a new quality metric for 3-d synthesized view assessment. *IEEE Journal of Selected Topics in Signal Processing*, 5(7):1332–1343, November 2011.

[BSCB00] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of ACM Conference on Computer Graphics (SIGGRAPH)*, pages 417–424, 2000.

[BVG$^+$11] B. Bartczak, P. Vandewalle, O. Grau, G. Briand, J. Fournier, P. Kerbiriou, M.Murdoch, M. M´uller, R. Goris, R. Koch, and R. van der

Vleuten. Display-independent 3d-tv production and delivery using the layered depth video format. *IEEE Transactions on Broadcasting*, 57(2):477–490, 2011.

[CHS13] S. Choi, B. Ham, and K. Sohn. Space-time hole filling with random walks in view extrapolation for 3d video. *IEEE Transactions on Image Processing*, 22(6):2429–2441, June 2013.

[CPT04] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13:1200–1212, 2004.

[CS01] T. F. Chan and J. Shen. Non-Texture Inpainting by Curvature-Driven Diffusions (CDD). *Journal of Visual Communication and Image Representation*, 12:436–449, 2001.

[CV95] J. E. Cutting and P. M. Vishton. Perceiving layout and knowing distances: the integration, relative potency and contextual use of different information about depth. *Handbook of perception and Cognition*, 5:69–117, 1995.

[DGK$^+$09] M. Domański, T. Grajek, K. Klimaszewski, M. Kurc, O. Stankiewicz, J. Stankowski, and K. Wegner. Poznañ multiview video test sequences and camera parameters. ISO/IEC JTC1/SC29/WG11 MPEG 2009/M17050, Xianl, China, October 2009.

[DS11] I. Daribo and H. Saito. A novel inpainting-based layered depth video for 3dtv. *IEEE Transactions on Broadcasting*, 57(2):533–541, 2011.

[DTPP07] I. Daribo, C. Tillier, and B. Pesquet-Popescu. Distance Dependent Depth Filtering in 3D Warping for 3DTV. In *IEEE 9th Workshop on Multimedia Signal Processing*, pages 312–315, October 2007.

[EL99] A. Efros and T. Leung. Texture Synthesis by Non-parametric Sampling. In *International Conference on Computer Vision*, pages 1033–1038, 1999.

[Feh04] C. Fehn. Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV. In *Proceeding. SPIE Stereoscopic Displays and Virtual Reality Systems XI*, pages 93–104, 2004.

[FWE13] A. Fernando, S. T. Worrall, and Ekmekcioglu E. *3DTV-Processing and Transmission of 3D Video Signals.* John Wiley & Sons, 2013.

[GAC$^+$11] A. Gotchev, G. Akar, T. Capin, D. Strohmeier, and A. Boev. Three-dimensional media for mobile devices. In *Proceedings of the IEEE*, volume 99, pages 708–741, April 2011.

[GM14] C. Guillemot and O. L. Meur. Image inpainting : Overview and recent advances. *IEEE Signal Processing Magazine*, 31(1):127–144, January 2014.

[GMG11] J. Gautier, O. L. Meur, and C. Guillemot. Depth-based image completion for view synthesis. In *3DTV conference*, pages 1–4, 2011.

[Han01] J. C. Handley. Comparitive analysis of Bradley-Terry and Thurstone-Mosteller paired comparison models for image quality assessment. In *IS and TS PICS Conference*, pages 108–112, 2001.

[Har01] P. Harrison. A Non-Hierarchical Procedure for Re-Synthesis of Complex Textures. In *Winter School of Computer Graphics Conf. Proc. (WSCG)*, pages 190–197, 2001.

[Hhi12] Test Model under Consideration for HEVC based 3D Video coding. ISO/IEC JTC1/SC29/WG11 MPEG2011/N12559, February 2012. San Jose, CA, USA.

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Univeristy Press, 2nd edition, 2004.

[IL02] H. Imaizumi and A. Luthra. *Three-Dimensional Television, Video, and Display Technologies.*, chapter Stereoscopic Video Compression Standard "MPEG-2 Multi-view Profile", pages 167–181. In [JO02], 2002.

[ISM05] W. A. IJsselsteijn, P. J. H. Seunties, and L. M. J. Meesters. *3D Video communication Algorithms, concepts and real-time systems in human centered communication*, chapter Human Factors of 3D displays, pages 219–233. In [SKS05], 2nd edition, 2005.

[ITU08] ITU-T recommendation p.910, subjective video quality assessment methods for multimedia applications. International Telecommunication Union, 2008.

[ITU12] ITU-R BT. 500-13, methodology for the subjective assessment of the quality of television pictures. International Telecommunication Union, January 2012.

[JF06] H. Jorke and M. Fritz. Infitec- a new stereoscopic visualization tool by wavelength multiplexing. Tecnical report Infitec, 2006.

[JGM11] V. Jantet, C. Guillemot, and L. Morin. Joint projection filling method for occlusion handling in depth-image-based rendering. In *3D Research*, 2011.

[JO02] B. Javidi and F. Okano. *Three-Dimensional Television, Video, and Display Technologies.* Springer Press Berlin, 2002.

[KES05] R. Koch and J. F. Evers-Senne. *3D Video communication Algorithms, concepts and real-time systems in human centered communication*, chapter View synthesis and rendering methods, pages 235–260. In [SKS05], 2nd edition, 2005.

[LKL$^+$11] H. Lim, Y. S. Kim, S. Lee, O. C., J.D.K. Kim, and C. Kim. Bi-layer inpainting for novel view synthesis. In *IEEE International Conference on Image Processing (ICIP)*, pages 1089–1092, September 2011.

[LR06] Y. G. Lee and J. B. Ra. Image distortion correction for lenticula misalignment in three-dimensional lenticular displays. *Optical Engineering*, 45(1):017007–017007–9, 2006.

[LWXD12] S. Li, R. Wang, J. Xie, and Y. Dong. Exemplar Image Inpainting by Means of Curvature-Driven Method. In *2012 International Conference on Computer Science and Electronics Engineering (ICCSEE)*, pages 326–329, 2012.

[MFW08] Y. Morvan, D. Farin, and P. H. N. De With. System architecture for free-viewpoint video 3D-TV. *IEEE Transactions Consumer Electronics*, 54(2):925–932, 2008.

[MFY$^+$09] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto. View generation with 3D warping using depth information for FTV. *Signal Processing; Image Communication*, 24(1-2):65–72, 2009.

[Mor09] Y. Morvan. Acquisition, Compression and Rendering of Depth and Texture for Multi-View Video. Technical report, Eindhoven University of Technology, 2009.

[MOS13a] S. M. Muddala, R. Olsson, and M. Sjöström. Depth-included curvature inpainting for disocclusion filling in view synthesis. *International Journal On Advances in Telecommunications*, 6(3 & 4):132–142, 2013.

[MOS13b] S. M. Muddala, R. Olsson, and M. Sjöström. Disocclusion handling using depth-based inpainting. In *The Fifth International Conferences on Advances in Multimedia(MMEDIA)*, April 2013.

[MSD$^+$08] K. Muller, A. Smolic, K. Dix, P. Merkle, P. Kauff, and T. Wiegand. View synthesis for advanced 3D video systems. *EURASIP Journal on Image and Video processing*, 2008:1–11, 2008.

[MSO12] S. M. Muddala, M. Sjöström, and R. Olsson. Edge-preserving depth-image-based rendering method. In *International Conference on 3D Imaging 2012 (IC3D)*, December 2012.

[MSO14] S. M. Muddala, M. Sjöström, and R. Olsson. Depth-based inpainting for disocclusion filling. In *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2014*, pages 1–4, July 2014.

[MSOT13] S. M. Muddala, M. Sjöström, R. Olsson, and S. Tourancheau. Edge-aided virtual view rendering for multiview video plus depth. In *Proceeding Society of Photo-Optical Instrumentation Engineers (SPIE)*, pages 86500E–86500E–7, 2013.

[Mus] Multimedia Scalable 3D for Europe. [online] `http://www.muscade.eu/overview.html`. 15 February 2015.

[MVi] Multiview 3D displays. [online] `www.dimenco.eu`. 20 March 2013.

[NPS08] Multiview video test sequence and camera parameters. ISO/IEC JTC1/SC29/WG11 MPEG2008/M15419, 2008. Archamps, France.

[Onu11] L. Onural. 3D video technologies: An overview in research trends. In *Society of Photo-Optical Instrumentation Engineers (SPIE)*, 2011.

[OYH09] K. J. Oh, S. Yea, and Y. S. Ho. Hole filling method using depth based inpaiting for view synthesis in free viewpoint television and 3-d video. In *PCS*, pages 1–4, 2009.

[Pas05] S. Pastoor. *3D Video communication Algorithms, concepts and real-time systems in human centered communication*, chapter 3d displays, pages 235–260. In [SKS05], 2nd edition, 2005.

[Phi] Autostereoscopic 3D. [online] `www.usa.philips.com/c/3d-autostereoscopic-series/303923/cat/en/professional`. 20 March 2013.

[PJO$^+$09] Y. K. Park, K. Jung, Y. Oh, J. K. Kim, G. Lee, H. Lee, K. Yun, N. Hur, and J. Kim. Depth-image-based rendering for 3DTV service over T-DMB. *Signal Processing: Image Communication*, 24:122–139, 2009.

[RMOdBaI$^+$02] A. Redert, C. Fehn M. O. de Beec and, W. A. Ijsselsteijn, M. Pollefeys, L. Van Gool, E. Ofek, I Sexton, and P. Surman. Advanced three-dimenstional television systems technologies. . In *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, pages 313–319, 2002.

[SC02] J. Shen and T. F. Chan. Mathematical Models for Local Nontexture Inpaintings. *SIAM Journal on Applied Mathematics*, 62:1019–1043, 2002.

[SGHS98] J. W. Shade, S. J. Gortler, L. W. He, and R. Szelisk. Layered depth images. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 231–242, 1998.

[SHKO11] M. Sjöström, P. Härdling, Linda S. Karlsson, and R. Olsson. Improved depth-image-based rendering algorithm. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4, 2011.

[SK00] H.-Y. Shum and S. B. Kang. Review of image-based rendering techniques. volume 4067, pages 93–104, 2000.

[SKS05] O. Schreer, P. Kauff, and T. Sikor. *3D Video communication Algorithms, concepts and real-time systems in human centered communication*. John Wiley & Sons, 2nd edition, 2005.

[SMM$^+$09a] A. Smolic, K. Muller, P. Merkle, P. Kauff, and T. Wiegand. Accommodation and convergence when looking at binocular 3D images. In *Picture Coding Symposium, 2009. PCS 2009*, pages 1–4, 2009.

[SMM$^+$09b] A. Smolic, K. Muller, P. Merkle, P. Kauff, and T. Wiegand. An overview of available and emerging 3d video formats and depth enhanced stereo as efficient generic solution. In *Picture Coding Symposium, 2009. PCS 2009*, pages 1–4, 2009.

[SOS13] S. Schwarz, R. Olsson, and M. Sjöström. Depth sensing for 3dtv: A survey. *IEEE MultiMedia*, 20(4):10–17, October 2013.

[SS02] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[SSO14] S. Schwarz, M. Sjöström, and R. Olsson. A weighted optimization approach to time-of-flight sensor fusion. *IEEE Transactions on Image Processing*, 23(1):214–225, January 2014.

[Tan06] M. Tanimoto. FTV (free viewpoint television) for 3d scene reproduction and creation. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, pages 172–172, June 2006.

[TAZ$^+$04] W. J. Tam, G. Alain, L. Zhang, T. Martin, and R. Renaud. Smoothing depth maps for improved steroscopic image quality. *Three-Dimensional TV, Video, and Display III*, 5599:162–172, 2004.

[Tel04] A. Telea. An image inpainting technique based on the fast marching method. *J. Graphics, GPU, Game Tools*, 9(1):23–34, 2004.

[Til11] R. F. Tiltmanl. How "stereoscopic" television is shown, November 2011.

[TLD07] Z. Tauber, Z. N. Li, and M. S. Drew. Review and preview: Disocclusion by inpainting for image-based rendering. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 37(4):527–540, 2007.

[TLLG09] D. Tian, P. L. Lai, P. Lopez, and C. Gomila. View synthesis techniques for 3d video. In *Proceeding Society of Photo-Optical Instrumentation Engineers (SPIE)*, volume 7443, pages 74430T–74430T–11, 2009.

[Tsc02] David Tschumperlé. *PDE's Based Regularization of Multivalued Images and Applications*. PhD thesis, Nice, 2002.

[Tsc06] D. Tschumperlé. Fast anisotropic smoothing of multi-valued images using curvature-preserving pde's. *International Journal of Computer Vision*, 68(1):65–82, 2006.

[UBH$^+$08] G. M. Um, G. Bang, N. Hur, J. Kim, and Y. S. Ho. 3d video test material of outdoor scene. ISO/IEC JTC1/SC29/WG11/M15371, April 2008.

[UCES11] H. Urey, K. V. Chellappan, E. Erden, and P. Surman. State of the art in stereoscopic and autostereoscopic displays. *Proceedings of the IEEE*, 99(4):540–555, April 2011.

[vsr10] Report on experimental framework for 3D video coding. ISO/IEC JTC1/SC29/WG11 MPEG2010/N11631, October 2010. Guangzhou, China.

[WBSS04] Z. Wang, A.C. Bovik, H.R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[Wei96] J. Weickert. Theoretical foundations of anisotropic diffusion in image processing. *Computing, Suppl*, 11:221–236, 1996.

[Wei98] J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart, 1998.

[Whe38] C. Wheatstone. On some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical Transactions*, 120:371–394, 1838.

[WMG13] D. Wolinski, O. L. Meur, and J. Gautier. 3d view synthesis with inter-view consistency. In *ACM Multimedia*, 2013.

[WSI07] Y. Wexler, E. Shechtman, and M. Irani. Space-Time Completion of Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 29(3):463–476, 2007.

[Zen86] S. Di Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, 33(1):116–125, January 1986.

[ZKU$^+$04] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics.*, 23(3):600–608, August 2004.

[ZT05] L. Zhang and W. J. Tam. Stereoscopic image generation based on depth images for 3D TV. *IEEE Transactions on Broadcasting*, 51(2):191–199, June 2005.

[ZwPSxZy07] L. Zhan-wei, A. Ping, L. Su-xing, and Z. Zhao-yang. Arbitrary view generation based on DIBR. In *Intelligent Signal Processing and Communication Systems*, pages 168–171, December 2007.

Biography

Suryanarayana Murthy Muddala was born on the 14th of August 1984 in Malakapalli, Andhra Pradesh, India. He received the degree of Master of Science in Electrical Engineering with emphasis on Signal Processing from Blekinge Institute of Technology, Sweden in 2009.

In 2011 he started his full time PhD studies at the Mid Sweden University in Sundsvall, Sweden. During his PhD studies he was involved in several projects with respect to 3D video capture and rendering. During his PhD studies he traveled within Europe as well as to the US to attend conferences in the field of 3D research. He also attended three summer schools funded by the European Cooperation in Science and Technology (COST) Action: 3D media and computational architecture, in Tampere, Finland 2012, Plenoptic capture, processing and reconstruction in Sundsvall, Sweden 2013, and 3D content creation, perception and interaction in Budapest, Hungary 2014.

Suryanarayana Murthy is currently pursuing a PhD degree in computer and system science at the Mid Sweden University. His research area is on 3D Video application, mainly focused on rendering, and his research interests are computer vision and image analysis.

Errata for
Free View rendering for 3D Video

Suryanarayana M. Muddala

Department of Information and Communication Systems
Mid Sweden University

ISBN 978-91-88025-30-2
ISSN 1652-893X
Sundsvall, Sweden
2015

- Page 24: in paragraph 2, line 2, "x-coordinates of corresponding points $\mathbf{m1}$ and $\mathbf{m1}$" should be "x-coordinates of corresponding points $\mathbf{m1}$ and $\mathbf{m2}$".
- Page 36: Eq. (4.5), "$D(\mathbf{p}) = \frac{\langle \nabla^{\perp} \cdot \mathbf{n_p} \rangle}{\alpha}$" should be "$D(\mathbf{p}) = \frac{\langle \nabla^{\perp} \mathbf{I_p} \cdot \mathbf{n_p} \rangle}{\alpha}$".
- Page 42: in Section 5.3.1, line2, "pixels inside the hole" should be "pixels inside the hole [Hhi12]".
- Page 50: in Figure 6.6 (b) legend, the symbol "+" should be added before interpolated value.
- Page 56: in caption Figure 6.8 "Poznan street" v4→v5 should be "Poznan street" v4→v3.
- Page 61: in paragraph 2, line 2, "described in Chapter 6" should be "described in Chapter 5".
- Page 87: in Section Occlusion mask generation, paragraph 2, after the sentence "where $\mathbf{f}_{\mathrm{v}}$, $\mathbf{p}_{\mathrm{v}}$ are projected pixels in $\mathbf{I}_{\mathrm{v}}$.", the following sentences should be added.

  "Note that the definition for DDP in Eq. (8.1) is presented within the context of a right warped image. It is straightforward to change Eq. (8.1) for a left warped image."
- Page 88: in paragraph 2, after the sentence "All the occlusion layers in the LDI are formed in this manner.", the following sentences should be added.

  "Note that the identified occlusions in the original view, derived using Eq. (8.4), only exactly correspond to the disocclusions in the virtual view for a parallel camera arrangement. For an arbitrary camera arrangement the correct information is obtained by inverse warping."