

Depth-guided Disocclusion Inpainting for Temporal Consistent Virtual View Synthesis

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Visual Computing

eingereicht von

Thomas Rittler, BSc

Matrikelnummer 0125728

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Mag.rer.nat. Dr.techn. Margrit Gelautz
Mitwirkung: Dipl.-Ing. Dr.techn. Florian Seitner

Wien, 10.03.2014

(Unterschrift Verfasser)

(Unterschrift Betreuung)

Depth-guided Disocclusion Inpainting for Temporal Consistent Virtual View Synthesis

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Visual Computing

by

Thomas Rittler, BSc

Registration Number 0125728

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao.Univ.Prof. Dipl.-Ing. Mag.rer.nat. Dr.techn. Margrit Gelautz

Assistance: Dipl.-Ing. Dr.techn. Florian Seitner

Vienna, 10.03.2014

(Signature of Author)

(Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Thomas Rittler, BSc
Eichbergstraße 3/6, 2372 Gießhübl

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Verfasser)

Acknowledgements

To begin with, I would like to thank my supervisor Margrit Gelautz who supported me in the last year of my study, established important contacts and made this thesis possible in the first place.

I would also like to express my sincere gratitude to Tom Wilson, and particularly to Florian Seitner for the provision of required hardware and software, as well as for his patience, motivation and enthusiasm. His guidance helped me in all the time of research for and writing of this thesis.

A special thanks goes to everyone who preferred a musty basement vault over a sultry summer afternoon and participated in my user study – I did really appreciate it.

Besides, I would like to express my sincere appreciation to Rainer Gross who was an important interlocutor and listener in good times as in bad.

My deepest and heartfelt thanks goes to my beloved girlfriend Ulli for her love, devotion and unstinting support. I enjoyed every second I could spend with you off of studying.

At last but not least, I would like to thank my family, especially my mother Christine and my brother Philipp, for their financial support and most notably for their love and continuous belief in me. My gratitude is beyond words!

This work was supported by the Technology Agency of the City of Vienna (ZIT) under the project PAINT3D.

Abstract

Depth Image Based Rendering (DIBR) has become an important tool in the field of free-viewpoint video and three-dimensional television as this technique allows to generate virtual views of a scene from a single view and its associated depth map. However, a common issue in DIBR are disocclusions, i.e. areas that are uncovered in the newly synthesized perspectives. This thesis addresses the problem of filling disocclusions in the context of stereoscopic images and three-dimensional videos. To synthesize visually plausible content in the disoccluded regions of the novel views, a texture-oriented inpainting method is presented based on the randomized correspondence algorithm PatchMatch of Barnes et al. As missing time-consistency results in disturbing flicker artifacts, our proposed inpainting approach exploits temporal information by propagating both patch correspondences and color information to subsequent frames. Moreover, the depth information available in the field of stereoscopic imaging is incorporated to avoid the interference of foreground color in the restored image background regions. To further account for artifacts caused by the image warping process as well as inaccuracies in the underlying depth map, several enhancements are introduced including unilateral dilation, adaptive patch sizes and a matching threshold. An extensive evaluation using objective quality assessment metrics and a subjective user study shows that the spatial and temporal enhancements of the proposed image and video completion framework clearly outperform existing inpainting tools such as “content-aware fill” from the professional graphics editing program Adobe Photoshop.

Kurzfassung

“Depth Image Based Rendering” ist zu einem wichtigen Werkzeug im Bereich der sogenannten “Free-Viewpoint”-Videos und des dreidimensionalen Fernsehens geworden, da dieses Verfahren, ausgehend von einem einzelnen Blickwinkel und der dazugehörigen Tiefenkarte, die Erzeugung virtueller Ansichten einer Szene ermöglicht. Eine häufig auftretende Problematik hierbei stellen jedoch Aufdeckungen dar, also jene Bereiche, die in den neu erzeugten Perspektiven freigelegt werden. Die vorliegende Diplomarbeit befasst sich mit dem Auffüllen dieser in stereoskopischen Bildern und dreidimensionalen Videos entstandenen Lücken. Um adäquate Bildinformation in den freigelegten Bereichen der neuen Ansichten zu generieren, wird eine texturbasierte Inpainting-Methode präsentiert, unter der Verwendung des randomisierten Korrespondenzalgorithmus PatchMatch. Da zeitlich inkonsistente Inpainting-Resultate von aufeinanderfolgenden Bildern zu störenden Flacker-Artefakten führen, wird in dem vorgestellten Bildvervollständigungsansatz temporale Information eingesetzt, wobei sowohl Patchkorrespondenzen, als auch Farbinformationen in nachfolgende Bilder propagiert werden. Des Weiteren wird die im Bereich der Stereoskopie vorhandene Tiefeninformationen einbezogen, um die Einmischung von Farbanteilen des Vordergrundes in die Inpainting-Ergebnisse zu unterdrücken. Um zusätzlich den Einfluss von Artefakten aufgrund von Ungenauigkeiten in den Tiefenkarten zu reduzieren, werden verschiedene Verbesserungen, wie unilaterale Dilatation, adaptive Patchgrößen und ein Übereinstimmungsschwellwert, vorgestellt. Eine umfangreiche Evaluierung mittels objektiver Qualitätsmetriken und einer subjektiven Benutzerstudie zeigt, dass die mit Hilfe der räumlichen und zeitlichen Erweiterungen erhaltenen Ergebnisse des vorgestellten Bild- und Videovervollständigungsansatzes jene bestehender Inpainting-Werkzeuge, wie das im professionellen Bildbearbeitungsprogramm Adobe Photoshop vorhandene “Inhaltssensitive Füllen”, an Qualität übertreffen.

Contents

List of Figures	xi
List of Tables	xii
Nomenclature	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Problem description	3
1.3 Aim of the work	4
1.4 Structure of the work	6
2 State of the Art	7
2.1 Image inpainting	7
2.1.1 Structural approaches	7
2.1.2 Exemplar-based approaches	9
2.2 Nearest-neighbor search	12
2.3 Video inpainting	13
2.4 Disocclusion filling	15
3 Algorithm and Implementation	17
3.1 PatchMatch	17
3.2 Hole filling using initial PatchMatch	19
3.3 Non-occupied target patches (“Blanks”)	21
3.4 Additional preprocessing steps	25
3.4.1 Single-sided mask dilation	25
3.4.2 Diffusion-based infilling of “small” holes	27
3.5 Adaptive patch sizes	29
3.6 Valid matching pixel threshold	30
3.7 Depth extension	33
3.8 Temporal extension	39
3.9 Additional algorithmic adaptations	42
3.9.1 Patch distance metric	42
3.9.2 Color space	44
	ix

3.9.3	Pixel color synthesis	45
3.10	Compared methods	47
3.10.1	Horizontal background replication	47
3.10.2	Adobe Photoshop: content-aware fill	48
4	Evaluation and Results	53
4.1	Database	53
4.2	Objective metrics	56
4.2.1	Still images	57
4.2.2	Video sequences	59
4.3	Subjective user study	61
4.3.1	Still images	63
4.3.2	Video sequences	69
4.4	Runtime analysis	71
5	Conclusion and Future Work	73
A	Listings of Evaluation Results	75
A.1	Objective metrics	76
A.2	Subjective user study	77
	Bibliography	81

List of Figures

1.1	Epipolar geometry of a rectified image pair	2
1.2	Disocclusions in virtual view synthesis	3
1.3	Depth and border holes	5
2.1	Isophotes	8
2.2	Structural inpainting example	9
2.3	Texture synthesis algorithm of Efros and Leung	10
2.4	Structure propagation according to Criminisi et al.	11
2.5	Mosaic image used by contour-based video inpainting	14
2.6	Space-time consistency according to Wexler et al.	14
3.1	Nearest-neighbor field	17
3.2	PatchMatch workflow	18
3.3	Exemplar-based inpainting	19
3.4	PatchMatch-based infilling workflow using onetime update	20
3.5	Inpainting system inputs	21
3.6	Holes and blanks (patch size = 21)	22
3.7	Holes and blanks (patch size = 51)	23
3.8	Zero overlap example	25
3.9	Single-sided mask dilation	26
3.10	Diffusion-based infilling of small holes	28
3.11	Adaptive patch sizes	29
3.12	Matching pixels threshold	31
3.13	Inpainting results using matching pixels threshold	32
3.14	Ghost shadowing artifacts	34
3.15	Depth-based inpainting	35
3.16	Scanline-based disparity map infilling	36
3.17	Depth-based inpainting enhanced	38
3.18	‘ANNF propagation’ workflow	40
3.19	‘Previous frame as reference’ workflow	40
3.20	Flickering artifacts	41
3.21	Inessentiality of spatial constraints	43
3.22	Pixel color synthesis	46

3.23	PatchMatch-based infilling workflow using iterative updates	49
3.24	Multiscale approach	49
3.25	Inpainting results of compared methods	51
4.1	Input images for evaluation	54
4.2	Sample frames of video sequences	55
4.3	Objective image quality assessment	58
4.4	Objective video quality assessment	60
4.5	Relative frame differential flicker results	61
4.6	Subjective user study	62
4.7	Pair comparison scores for still images	64
4.8	Inpainting results: DPM versus PM	65
4.9	Inpainting results: DPM versus HBR	66
4.10	Inpainting results: DPM versus APM	67
4.11	Inpainting results: DPM versus CAF	68
4.12	Pair comparison scores for video sequences	70
4.13	Visual disturbances due to warping artifacts	71
4.14	Pair comparison score versus runtime	72
4.15	Runtime distribution	72
A.1	Pairwise comparison of image inpainting approaches	79
A.2	Pairwise comparison of video inpainting approaches	80

List of Tables

3.1	Potential patch information levels	24
3.2	Two kernels for isotropic diffusion	27
4.1	Specification of video sequences	55
4.2	Abbreviation of evaluated methods	56
A.1	Objective metric scores of still images	76
A.2	Objective metric scores of video sequences	76
A.3	Full pair comparison scores of still images	77
A.4	Full pair comparison scores of video sequences	77
A.5	Shortened pair comparison scores of still images	78
A.6	Shortened pair comparison scores of video sequences	78

Nomenclature

2D	Two-Dimensional
3D	Three-Dimensional
3DTV	Three-Dimensional TeleVision
ANN	Approximate Nearest-Neighbor
ANNF	Approximate Nearest-Neighbor Field
ARGB	Alpha Red Green Blue
CDD	Curvature Driven Diffusion
CIE	Commission Internationale de l'Éclairage
CSH	Coherency Sensitive Hashing
DIBR	Depth Image Based Rendering
FMM	Fast Marching Method
FTV	Free-viewpoint TeleVision
FVV	Free-Viewpoint Video
GMM	Gaussian Mixture Model
GT	Ground Truth
HD	High Definition
HSV	Hue Saturation Value
ITU	International Telecommunication Union
LCD	Liquid Crystal Display
LDV	Layered Depth Video

LED	Light-Emitting Diode
LSH	Locality Sensitive Hashing
MRF	Markov Random Field
MSSIM	Mean Structural SIMilarity
MVD	Multiview Video-plus-Depth
NN	Nearest-Neighbor
NNF	Nearest-Neighbor Field
NNS	Nearest-Neighbor Search
PC	Pair Comparison
PCA	Principal Component Analysis
PDE	Partial Differential Equation
PSNR	Peak Signal-to-Noise Ratio
RFDF	Relative Frame Differential Flicker
RGB	Red Green Blue
RMSE	Root Mean Square Error
SAD	Sum of Absolute Differences
SSD	Sum of Squared Differences
SSIM	Structural SIMilarity
TSVQ	Tree Structured Vector Quantization
TV	Total Variational

Introduction

1.1 Motivation

Recently, the broad acceptance and the commercial success of movies like “*Avatar*” have paved the way for *Three-Dimensional* (3D) visual technologies – and consequently for 3D devices including LCD/LED displays, cameras, laptops, tablets, mobile phones and portable game consoles – to the consumer market. Moreover, *Three-Dimensional TeleVision* (3DTV) has become a highly active area of research in the computer vision community over the last years, as it is believed to be the future of television broadcasting that brings a more life-like and visually immersive home entertainment experience to the user [CLL11; SA12].

According to the binocular human visual system, traditional stereoscopic frameworks deliver distinct image contents to each eye of the observer with the assistance of e.g. anaglyph, shutter or polarized glasses. Particularly, the individual images seen by the left and right eye are combined by the brain to form a single 3D percept, where the difference in image location of an object between the left and the right view, named as *parallax* or *disparity*, is utilized to extract depth information from the *Two-Dimensional* (2D) images. However, in general these systems only provide a single perspective onto the scene. Thus, especially *Free-viewpoint TeleVision* (FTV) and *Free-Viewpoint Video* (FVV), which allow users to interactively navigate through a scene in order to choose a different perspective as well as autostereoscopic displays that provide 3D perception from distinct viewpoints to multiple observers without the need for special glasses, have increasingly attracted interest [DP10; AK12].

However, capturing and broadcasting arbitrary viewpoints for e.g. FTV would require an unrealistic number of cameras, extremely complex coding, and expensive processors, since each individual perspective in a stereoscopic video requires two input streams according to the left and right camera view [SA12]. Hence, a common approach based on a standard two-camera stereo setup is to synthesize virtual viewpoints using *Depth Image Based Rendering* (DIBR). Particularly, starting from a single reference view and its corresponding depth or disparity map – named as *2D-plus-depth* or *video-plus-depth* representation – additional perspectives are rendered through 3D image warping and intermediate angles are interpolated afterwards. This

enables the synthesis of a flexible number of views, which allows to account for advances in 3D display technologies.

Additionally, the conversion of available 2D content for rerelease in 3D is a highly important topic for content providers and for the success of 3D videos in general. It naturally completely relies on virtual view synthesis of a second view given the original 2D video [SKK+11]. Several formats of 3D video data representations like *Layered Depth Video* (LDV) [MSD+08] and *Multiview Video-plus-Depth* (MVD) [SMD+08] have been proposed that provide one full central view and residual texture and depth information to generate additional side perspectives by view synthesis. Consequently, this leads to only a limited bandwidth increase for the transmission of 3D videos compared to 2D content.

A simple approach to calculate the new viewpoints without requiring 3D rendering or 3D representation is the so-called *disparity morphing* as proposed by Alessandrini et al. [ABP09]. As can be seen in Figure 1.1, if the images are aligned to be coplanar, corresponding pixels lie on the same scanline in both images. Hence, assuming that, due to image rectification based on epipolar constraints, solely horizontal parallax information is used to create the 3D impression, the aim is to shift the initial horizontal pixel position p_0 proportionally to its disparity (or inverse depth value¹):

$$p_s = p_0 + s(d_{p_0} - \delta) \quad , \quad (1.1)$$

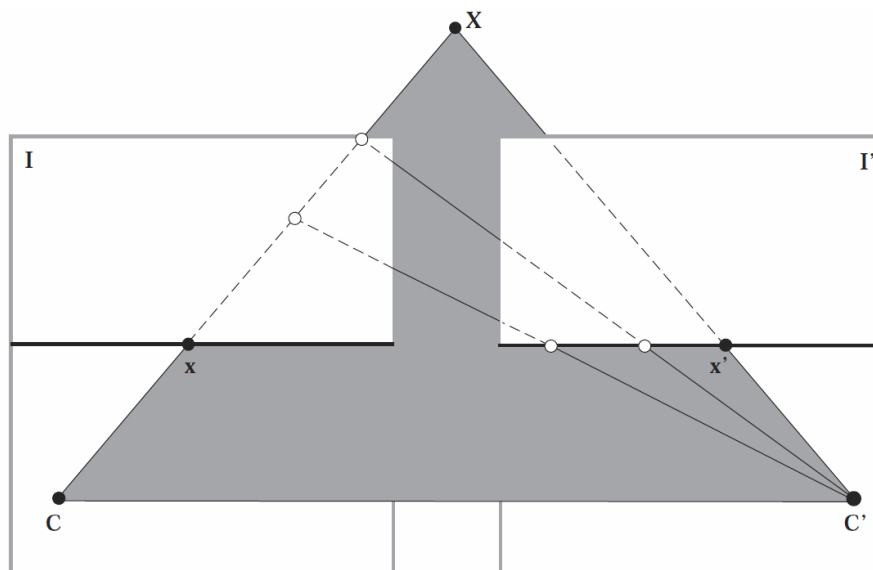


Figure 1.1: Epipolar geometry of a rectified image pair \mathbf{I} and \mathbf{I}' , where \mathbf{C} and \mathbf{C}' represent the centers of projection of the two cameras and \mathbf{x} and \mathbf{x}' are the projections of the object point \mathbf{X} onto the image planes. Since all epipolar lines are horizontal and parallel, corresponding points lie on the same scanline [B11].

¹Note that disparity and depth are inversely proportional.

where d_{p_0} is the disparity of the pixel p_0 , s a control parameter corresponding to the distance between virtual camera positions and δ the convergence point control parameter, which determines the amount of the disparity offset. As stated in [ABP09], this produces the following stereoscopic effects:

1. $d > \delta$: Point appears *behind* the screen plane
2. $d = \delta$: Point appears *on* the screen plane
3. $d < \delta$: Point appears *in front* of the screen plane

The color value at position p_s in the novel view is equivalent to the color value at the corresponding position p_0 in the original view.

1.2 Problem description

Two related problems that arise when generating virtual views are *occlusions* and *disocclusions*. Occlusions occur when different pixel positions p_0, \dots, p_n in the reference view are projected onto the same position p_s in the virtual view. On the other hand, disocclusions denote areas that are uncovered in the newly synthesized perspective as schematically illustrated in Figure 1.2. As can be seen, in a standard two-camera stereo setup, disocclusions appear particularly at the synthesis of extrapolated viewpoints, since missing information in intermediate views, i.e. along

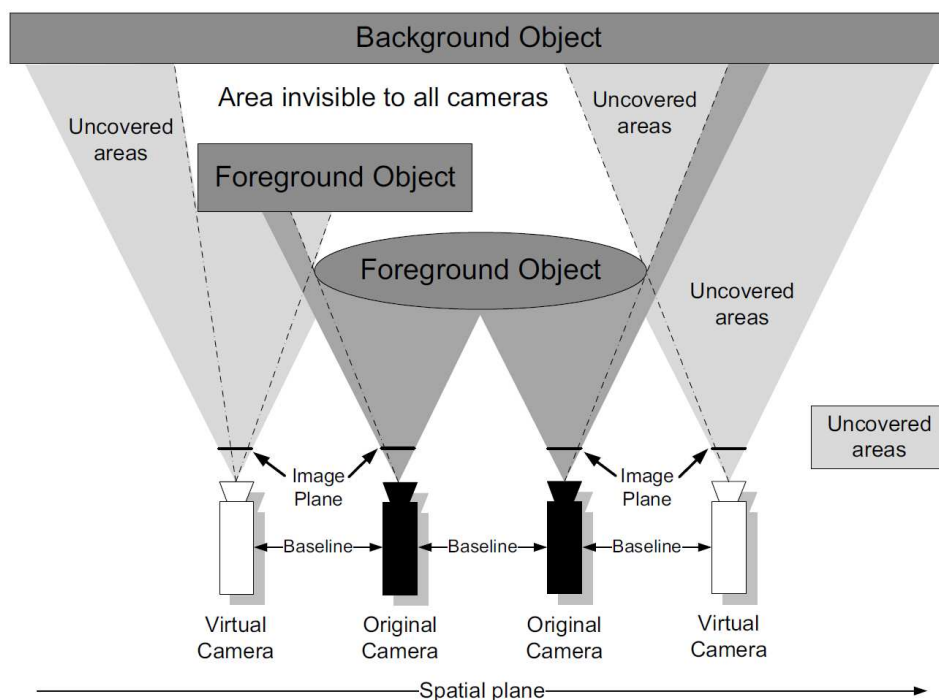


Figure 1.2: Uncovered areas (*disocclusions*) due to extrapolated virtual camera views [KWD+12].

the baseline axis *between* the left and right camera, can be interpolated by fusing available information from both camera images according to depth reliability maps [LLL+12].

Hence, while occlusions can be simply resolved by comparing the disparity values of overlapping pixels² – as pixels of foreground objects occlude those of the background – disocclusions pose a crucial problem in the wake of virtual view synthesis. In particular, there are two main reasons for the appearance of disocclusions:

1. Areas that are occluded in the original view and have become visible in the virtual view (“*depth holes*”)
2. Areas that become visible in the altered field of view of the virtual camera (“*border holes*”)

Figure 1.3 shows the occurrence of border and depth holes in an example image. As can be seen by the gap along the boundary of the character, disocclusions primarily occur at sharp depth discontinuities between foreground and background, since the virtual view allows to “look behind” the foreground object to see information in the background that is occluded in the original view. Besides disocclusions, which typically induce larger uncovered areas relative to the length of the baseline, inaccuracies in the depth estimation process as well as pixel position quantization after the warping step lead to additional cracks and small holes in the rendered images (“*warping artifacts*”) [SMD+08].

1.3 Aim of the work

To avoid the appearance of disocclusions – and consequently that of holes – in virtual view synthesis, one common approach is to preprocess the depth maps. In particular, prior to the warping stage, a low-pass filter is applied to smooth depth data across the edges and thus lowering the depth gradients in the virtual view [ZT05; VTS06]. However, smooth transitions in the depth map introduce distortion in the scene geometry and diminish the depth resolution contained in the rendered stereoscopic view, which results in an impairment of the overall 3D impression.

An alternative approach is to fill in the newly exposed areas in the synthesized views using digital inpainting techniques. In general, *image inpainting*, also known as *image completion*, is the recovery of missing or corrupted parts of an image in a given target region Ω , so that the reconstructed frame looks natural. It has become a standard tool in digital photography for image retouching and intensive research is under way to convert digital inpainting into a key tool for video and 3D cinema post-production [C11]. Additionally, as displays with 50 views and more are expected in a few years which require larger baseline rendering, digital inpainting has turned out to be a promising approach to remedy the disocclusion problem [NKD+11].

In this thesis, the problem of filling disocclusions in the field of stereoscopic images and 3D videos is addressed. To synthesize visually plausible content in the disoccluded regions of the virtual views, the usage of *PatchMatch* [BSF+09; BSG+10] in a DIBR framework is investigated, which depicts a randomized algorithm for quickly finding approximate nearest-neighbor

²The effect of this process is comparable to a *Z-buffer* in the field of computer graphics that keeps the pixels of smallest depth (or greatest disparity) when an occlusion appears [ABP09].



(a) Original view



(b) Virtual view

Figure 1.3: Pseudo-3D representation of a sample image showing the occurrence of depth (marked in red) and border holes (marked in blue) as the viewpoint changes.

correspondences between image patches. Based on a standard 2D-plus-depth setup, i.e. a single view and its associated disparity map³, an exemplar-based inpainting approach is proposed, which incorporates the supplementary depth information that is available in a stereoscopic environment. Besides the usage of spatial information, also the temporal component of videos is considered to avoid flickering artifacts by providing consistent inpainting results.

1.4 Structure of the work

The rest of this thesis is organized as follows:

- **Chapter 2** provides a comprehensive literature survey on digital image and video completion approaches including different techniques for the determination of nearest-neighbor correspondences, which denotes the core element of every exemplar-based inpainting method. Additionally, the related works considering the problem of filling disocclusions in a DIBR framework are reviewed.
- **Chapter 3** introduces the proposed hole filling approach. Starting from the original Patch-Match algorithm of Barnes et al., which constitutes the foundation of the proposed framework, the drawbacks of their patch matching method are analyzed and several enhancements are presented. These enhancements comprise additional preprocessing steps, adaptive patch sizes, a matching threshold, as well as depth-based and temporal extensions. The two inpainting approaches used for evaluation – Adobe[®]'s *content-aware fill* and *horizontal background replication* – are elucidated in addition.
- **Chapter 4** presents the evaluation results. After the chosen dataset and the implementation details are discussed, the different enhancement stages of the proposed inpainting framework are compared to the aforementioned image completion approaches. To evaluate both the depth-based and the temporal extensions of the presented framework, the inpainting methods are applied to still images and video sequences. Moreover, the evaluation of the inpainting techniques is conducted with objective quality metrics as well as a subjective user study.
- **Chapter 5** concludes this thesis and provides an outlook on future work.

³In this thesis, the disparity maps are calculated by an adapted version of the stereo correspondence algorithm proposed in [BG08].

State of the Art

In the field of computer vision, *inpainting* – also referred to as *infilling* or *completion* – is the process of filling missing data into a designated region of a still or video image, in such a manner that an observer cannot detect the restored parts [BBS01]. This chapter presents an overview of significant scientific papers in the area of image and video completion. Furthermore, the related works considering the problem of filling disocclusion in a DIBR framework are addressed, together with the nearest-neighbor correspondence search, which denotes an essential part of exemplar-based infilling approaches.

2.1 Image inpainting

Most inpainting methods found in the literature can be classified into two groups: *geometry-* and *texture-oriented* methods [AFC+11]. Geometrical – also named as *structural* – inpainting reconstructs using prior assumptions about the smoothness of structures in missing regions and boundary conditions, while textural – also referred to as *exemplar-based* – inpainting considers only available data from texture exemplars or other templates in non-missing regions [DP10]. Several notable works of these two categories are subsequently discussed.

2.1.1 Structural approaches

In 2000, Bertalmio et al. [BSC+00] first coined the terminology *image inpainting* in the context of digital image processing. Attempting to imitate the basic techniques used by professional paintings conservators, the main idea behind their algorithm is to continue the lines and structures that surround the vacant region into the hole. In particular, both the geometric (gradient direction) and the photometric (gray values) information is propagated inside the target region along the direction of the isophotes, which correspond to lines of equal gray values as illustrated in Figure 2.1. For an image I , this is achieved by numerically solving the following high-order Partial Differential Equation (PDE) (t is an artificial time marching parameter):

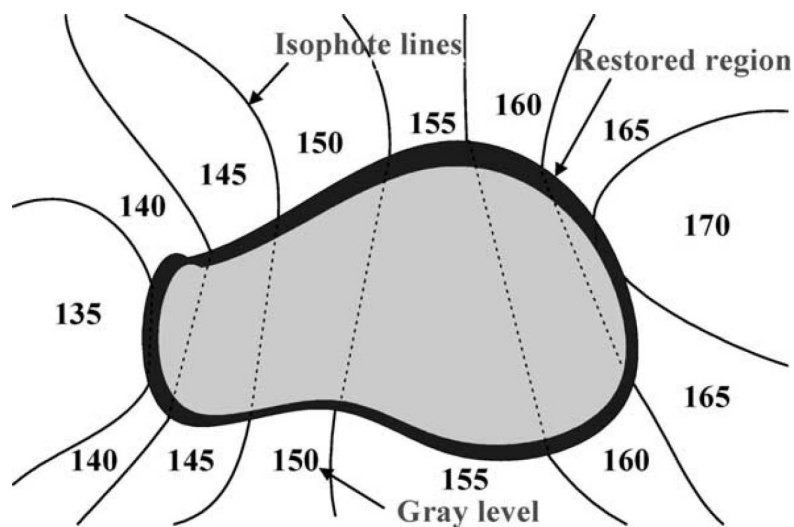


Figure 2.1: Example of a possible connection of isophotes at the restored region [KK03].

$$\frac{\partial I}{\partial t} = \nabla(\Delta I) \cdot \nabla^\perp, \quad (2.1)$$

where ∇ , Δ , and ∇^\perp denote the gradient, Laplacian and orthogonal-gradient (isophote direction)¹, respectively. At steady state $\frac{\partial I}{\partial t} = 0$, ΔI is constant along the direction ∇^\perp of the isophotes, thereby achieving a smooth continuation of the Laplacian inside the region to be inpainted [BVS+03]. In [BBS01], the relationship of the above equation with classical fluid dynamics is shown and a different flow according to the Navier-Stokes equation is presented to achieve the steady state.

Inspired by the work of Bertalmio et al., Chan and Shen [CS00] proposed the *Total Variational* (TV) model using an Euler-Lagrange equation coupled with anisotropic diffusion [PM90] to maintain the isophotes direction. The major drawback of the TV inpainting model is that it does not connect broken edges when the disconnected remaining parts are separated far apart by the inpainting domain. Thus, to allow inpainting to proceed over larger regions, Chan and Shen [CS01] extended the TV model to *Curvature Driven Diffusion* (CDD), which takes the geometric information of isophotes into account when defining the strength of the diffusion process.

However, all of the above mentioned inpainting algorithms are very time consuming and fast numerical implementations are difficult to achieve [CS01]. A notable improvement of PDE-based infilling approaches was introduced by Telea [T04], who uses the *Fast Marching Method* (FMM) [S95] to propagate the image information. Moreover, Oliveira et al. [OBM+01] proposed a fast convolution-based inpainting algorithm according to isotropic diffusion, where the regions to be filled are repetitively convolved with a predefined kernel. This method is employed as an additional processing step in the proposed image completion framework and will be discussed in detail in Section 3.4.2.

¹Note that the isophote direction ∇^\perp indicates the direction of the smallest change.

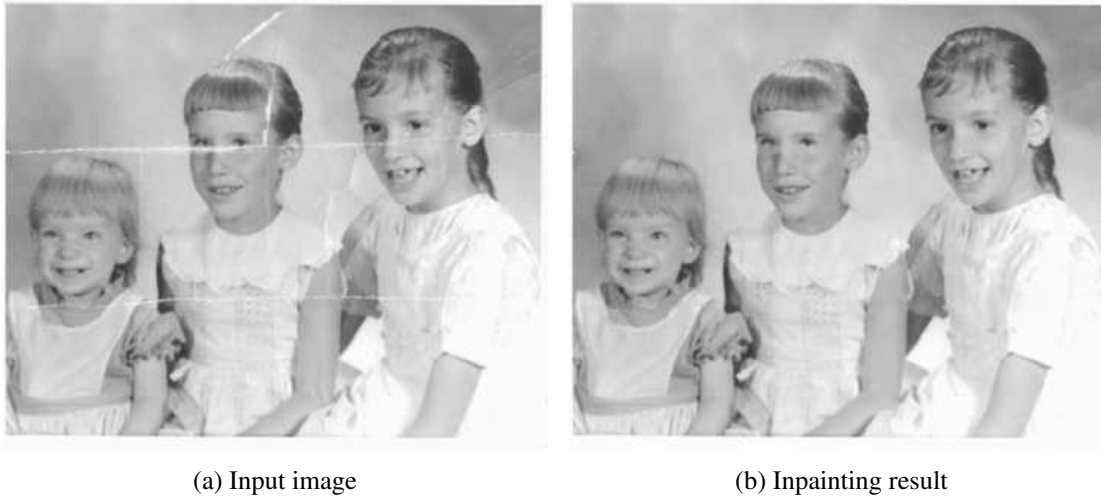


Figure 2.2: Restoration of an old photograph using the structural inpainting approach of Bertalmio et al. [BSC+00].

Since structural inpainting approaches are mainly designed for applications such as image denoising and restoration to e.g. remove overlaid text or cracks of damaged photographs, as shown in Figure 2.2, these methods are suitable for completing small, non-textured target regions. Thus, another drawback of these PDE-based algorithms is that they produce heavily blurred inpainting results caused by the diffusion process as the gaps become larger or are situated in highly textured areas, where the restoration of linear structures fails.

2.1.2 Exemplar-based approaches

Geometry-oriented methods are local in the sense that the associated PDEs only involve interactions between neighboring pixels on the image grid. An implication of this is that among all the data available in the image, they only use that around the boundary of the inpainting domain [AFC+11]. In contrast, texture-oriented inpainting approaches synthesize the visual information in the missing regions using the best matching *windows* – also named as *patches*, *exemplars* or *fragments* – from the remaining known portions of the entire image. The similarity of these patches is measured by certain patch distance metrics.

In the pioneer work of Efros and Leung [EL99], texture is modeled as a *Markov Random Field* (MRF) assuming that the probability distribution of brightness values for a pixel given the brightness values of its spatial neighborhood is independent of the rest of the image. The neighborhood is defined as a squared window around that patch, and corresponding samples of the desired texture are directly employed to perform the synthesis. However, a major deficit of their method is its computational cost as the texture synthesis is done pixel-by-pixel. An overview of Efros and Leung’s algorithm is given in Figure 2.3. Although their method was initially intended for texture synthesis, it has laid the foundation of several exemplar-based approaches by exploiting the self-similarity characteristic of natural images and has proven most effective for

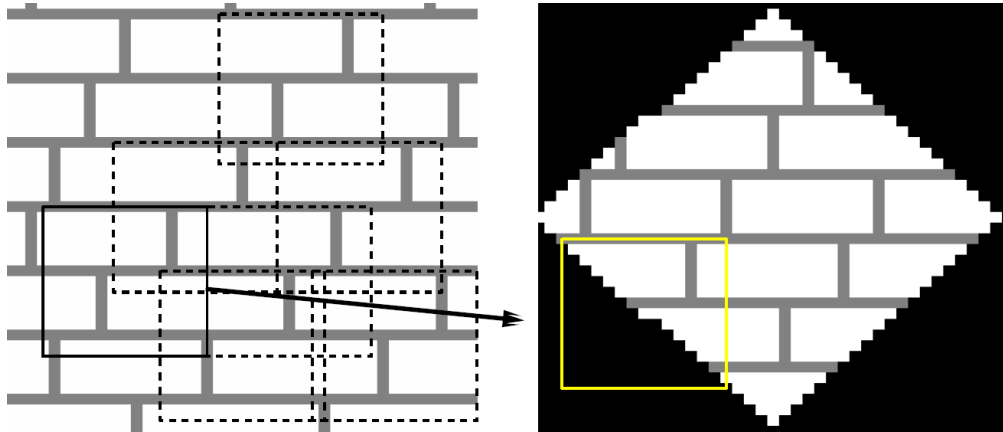


Figure 2.3: Algorithm overview of Efros and Leung. Given a sample texture image (left), a new image is being synthesized one pixel at a time (right). To synthesize a pixel, the algorithm first finds all neighborhoods in the sample image (boxes on the left) that are similar to the pixel’s neighborhood (box on the right) and then randomly chooses one neighborhood and takes its center to be the newly synthesized pixel [EL99].

the inpainting problem [C11].

For instance, Jia and Tang [JT03] segmented an image into several regions based on its color texture features and then inpainted each region individually. Given a neighborhood window of size $n \times n$ centered at a pixel p , this patch is transformed into a stick tensor by producing a feature vector of dimension $N = n \cdot n + 1$ in lexicographical ordering. The matching between corresponding feature vectors is translated into tensor voting for a straight line² in the N -dimensional space to account for color consistency. Drori et al. [DCY03] proposed a fragment-based image inpainting algorithm that iteratively adds details to the target region by composing adaptive fragments. A confidence map is used for comparing and selecting pairs of circular fragments, which determines how much confidence to place in image information at each pixel as new information is generated during the course of the algorithm. However, the computation time for these algorithms is impracticable, e.g. for the latter approach it takes more than two hours for a 384×256 image on a 2.4 GHz processor to accomplish the completion process [DCY03].

Exemplar-based methods provide impressive results in recovering textures and repetitive structures when enough samples are provided to synthesize the desired image, but their ability to recreate the geometry is limited [AFC+11]. Hence, Criminisi et al. [CPT03] presented a novel algorithm that builds on the work of Bertalmio et al. [BVS+03] by combining the strengths of both PDE-based image inpainting and non-parametric texture synthesis. Moreover, they demonstrated that the quality of the inpainting result is highly influenced by the order in which the completion is processed. In contrast to the default favorite “onion peel” method where the target region is synthesized in concentric layers from the outside inward, Criminisi et al. formulated the priority $P(\mathbf{p})$ of a patch $\Psi_{\mathbf{p}}$ centered at a point \mathbf{p} , where \mathbf{p} is located at the hole boundary $\delta\Omega$, as the product of two terms:

²This line defines a family of hyperplanes which are voted for by N -dimensional tensor voting.

$$P(\mathbf{p}) = C(\mathbf{p}) \cdot D(\mathbf{p}) \quad . \quad (2.2)$$

$C(\mathbf{p})$ is called the *confidence* term that indicates the reliability of the current patch, and $D(\mathbf{p})$ denotes the *data* term that gives special priority to the isophote direction. These are defined as follows:

$$C(\mathbf{p}) = \frac{1}{|\Psi_{\mathbf{p}}|} \sum_{\mathbf{q} \in \Psi_{\mathbf{p}} \cap \Phi} C(\mathbf{q}) \quad , \quad D(\mathbf{p}) = \frac{\langle \nabla I_{\mathbf{p}}^{\perp}, \mathbf{n}_{\mathbf{p}} \rangle}{\alpha} \quad , \quad (2.3)$$

where $|\Psi_{\mathbf{p}}|$ is the area of $\Psi_{\mathbf{p}}$ (i.e. the number of pixels), α is a normalization factor (e.g. $\alpha = 255$ for a typical gray-level image), $\mathbf{n}_{\mathbf{p}}$ is a unit vector orthogonal to the boundary $\delta\Omega$ in the point \mathbf{p} and $\nabla^{\perp} = (-\partial_y, \partial_x)$ is the direction of the isophote. During initialization, the confidence $C(\mathbf{q})$ of valid pixels $\mathbf{q} \in \Phi$ is set to 1. Once all priorities on the hole boundary have been computed and the patch $\Psi_{\hat{\mathbf{p}}}$ with the highest priority is found, the exemplar in the source region which is most similar to $\Psi_{\hat{\mathbf{p}}}$ under a given patch distance metric is determined. The corresponding source patch is then copied into the position occupied by $\Psi_{\hat{\mathbf{p}}}$ and the confidence $C(\mathbf{p})$ in this area is updated.

Figure 2.4 illustrates these algorithmic steps, which are repeated until the hole is completely filled. Since entire patches are copied instead of single pixels, this method attained a considerable speed-up compared to previous exemplar-based approaches, but the exhaustive search to determine the best candidate exemplars is still computationally expensive. Another drawback is the need to manually select the patch size, which is an important aspect that contributes to the overall quality of the inpainting result. The patch size must be chosen carefully to reflect the underlying characteristic of the image [M10], as too small patches are not able to capture

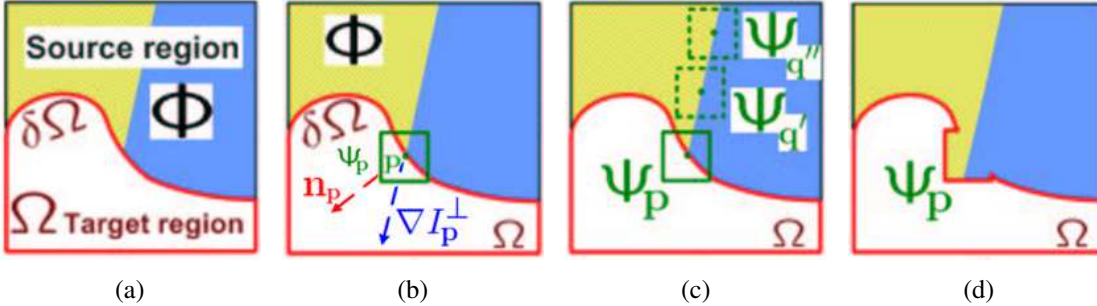


Figure 2.4: Structure propagation by exemplar-based texture synthesis. (a) Original image, with the target region Ω , its contour $\delta\Omega$ and the source region Φ clearly marked. (b) We want to synthesize the area delimited by the patch Ψ_p centered on the point $p \in \delta\Omega$, \mathbf{n}_p is the normal to $\delta\Omega$ and ∇I_p^{\perp} is the isophote. (c) The most likely candidate matches for Ψ_p lie along the boundary between the two textures in the source region, e.g. $\Psi_{q'}$ and $\Psi_{q''}$. The best matching patch in the candidates set has been copied into the position occupied by Ψ_p , thus achieving partial filling of Ω . The target region Ω has now shrunk and its boundary has assumed a different shape [CPT03].

the texture, whereas to large fragments may exhibit redundant information. Moreover, Criminisi et al. noted that their algorithm encounters some difficulties in reconstructing curved structures.

Comprehensive surveys of digital image inpainting techniques are given in [F08; SSB11; JJ12; MV12; RPM+13].

2.2 Nearest-neighbor search

The core element of every exemplar-based inpainting approach – as well as the most time consuming processing step – is the search for correspondences of patches in the target region and their most similar exemplars in the remaining image portions, known as the *Nearest-Neighbor Search* (NNS). Since the number of patches in an image is in the millions, so-called *Approximate Nearest-Neighbor* (ANN) algorithms are developed that exhibit only a minor loss in accuracy. To further increase the efficiency of NNS algorithms, various methods have been proposed using tree-based data structures, such as *kd-trees* [B75] and *Tree Structured Vector Quantization* (TSVQ) [WL00], which are often coupled with dimensionality reduction methods like *Principal Component Analysis* (PCA). Although these approaches have the advantage to reduce the number of search candidates due to their tree-structured organization of the information, they may not be appropriate for large-scale data, since the construction of such data structures typically takes significant time and requires large memory space [KR02; HHA12].

A different strategy was pursued by Ashikhmin [A01], who introduced a local propagation technique based on the coherent structure of images. In his texture synthesis process, the search space for a patch was limited to the source locations of its neighbors in the exemplar texture. This method was extended to *k-coherence search* in [TZL02] by caching the k nearest-neighbors of each patch as a preprocessing step. Although this accelerates the search phase, k -coherence still requires a full NNS for all pixels in the input, and has only been demonstrated in the context of texture synthesis [BSF+09]. Inspired by these works, Barnes et al. [BSF+09] proposed a novel approach, termed *PatchMatch*, that outperformed predominant tree-structured approaches by at least one order of magnitude, establishing image editing applications to run at interactive rates. Their algorithm and its generalized version [BSG+10] exploit the natural structure of images by propagating good matches to nearby patches. Additionally, a subsequent random search step is employed to prevent the algorithm from being trapped in local minima. PatchMatch constitutes the basis of the proposed image completion framework and is discussed in detail in Section 3.1.

It is worth noting that recently several adaptations of the PatchMatch algorithm have been presented. In [RB12], a low-dimensional version thereof was introduced that reduces the dimensions of an image patch to a set of low-level features. Korman and Avidan proposed *Coherency Sensitive Hashing* (CSH) [KA11], where the random search step of PatchMatch was replaced by a hashing scheme based on the same principle as *Locality Sensitive Hashing* (LSH) [IM98]. Olonetsky and Avidan [OA12] combined the PatchMatch approach with *kd-trees* and decreased the computational cost in the propagation stage by exploiting the overlap of adjacent patches using integral images [VJ01].

2.3 Video inpainting

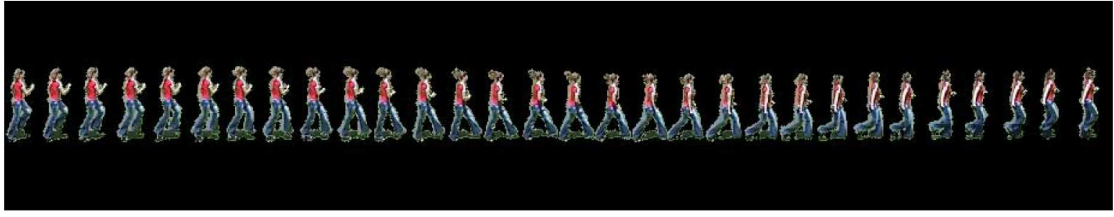
A straightforward extension of digital image inpainting to the field of video inpainting is to treat the underlying video data as a set of distinct frames and apply the existing image inpainting algorithms to them individually [M10]. For instance, Bertalmio et al. [BBS01] performed their PDE-based inpainting approach on a frame-wise basis by continuously propagating isophote lines from the hole boundaries, as elucidated in Section 2.1.1.

However, this strategy of employing image inpainting algorithms directly to video data does not take full advantage of the high temporal correlation that exists in video sequences. Consequently, this will cause undesirable artifacts in the inpainting result as only the spatial consistencies within a frame are considered. Since video inpainting approaches typically need to cover a considerably greater number of pixels and also the search space for finding patch correspondences in exemplar-based approaches is substantially larger, video inpainting is often considered as a much more challenging problem compared to image inpainting. Although the amount of work proposed in video completion is comparatively smaller than that of image inpainting, several notable methods have been presented in the recent years [M10; APS12].

Patwardhan et al. [PSB05] adapted the priority-driven image inpainting technique of Criminisi et al. to video sequences. After segmenting the input sequence into moving foreground objects and static background using optical flow, holes in the background layer are completed by directly copying available temporal information in the unimpaired frames. The remaining vacant regions of the background are then filled in by finding the highest priority location and propagating the best matching patch to all frames in order to maintain a consistent background throughout the sequence. Moreover, missing information of the foreground object is completed by copying only the moving part of the corresponding patch according to a priority based approach. As stated in [M10], while being effective in completing weakly-structured regions, this method, like its image inpainting counterpart, is susceptible to providing inconsistent infilling results already at a small number of incorrect patch correspondences. Additionally, their solution is not suitable for arbitrary camera motion.

In [GS11], Ghanbari and Soryani presented a contour-based video inpainting approach. Similar to Patwardhan et al. [PSB05], the moving foreground and the stationary background are first separated, but in this method via background subtraction according to a Gaussian Mixture Model (GMM) [WAD+97]. To complete the moving foreground, bounding boxes of the non-occluded foreground object are determined in all frames, which are then aligned to acquire a mosaic image, as shown in Figure 2.5. To select the most suitable patch, which is considered to be large enough to enclose an object entirely, the absolute difference between the contours of all complete patches and the object right before the hole is calculated. Finally, the inpainting is done using the exemplar-based approach of Criminisi et al. However, this video inpainting method assumes a static background and periodic motion of the foreground object without any changes in scale.

Wexler et al. [WSI07] proposed a prominent video completion algorithm that applies a global optimization strategy. In their work, 3-dimensional spatio-temporal patches are sampled from different parts of the video sequence to complete the missing video portions. In particular, global consistency is obtained by enforcing an objective function which demands that all overlapping



(a)

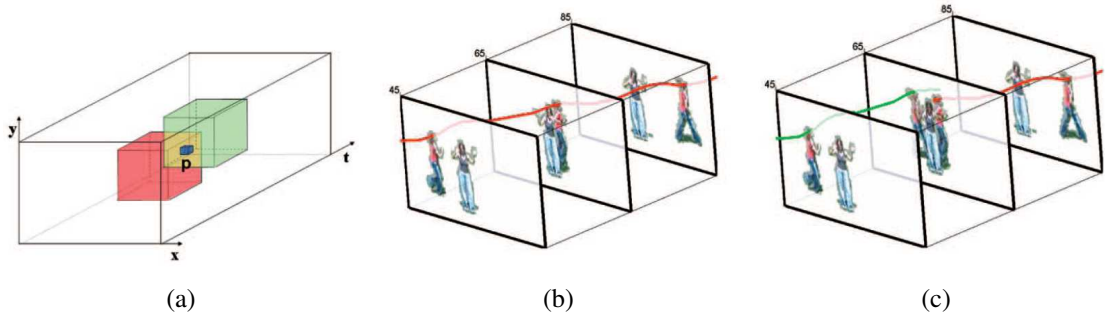


(b)

Figure 2.5: Mosaic image of (a) non-occluded foreground objects and (b) corresponding contours [GS11].

patches in the vicinity of a space-time point agree on its color value, as illustrated in Figure 2.6. Though both spatial and temporal information are handled simultaneously, the multi-scale nature of the solution may lead to blurry results [APS12; AD13]. As their approach has also been partially included in the *content-aware fill* of the commercial image editing program Adobe[®] Photoshop[®], which is used for evaluation in this thesis, the non-temporal aspects of this method will be discussed in detail in Section 3.10.2.

Besides patch-based methods, the second type of digital video inpainting algorithms is denoted as *object-based* approaches. For instance, Jia et al. [JTW06] employ a user-assisted video



(a)

(b)

(c)

Figure 2.6: Local and global space-time consistency. (a) Enforcement of the global objective function of Equation 3.12 requires coherence of all space-time patches encompassing the point p . (b) Such coherence leads to a globally correct solution. The true trajectory of the moving object is marked in red and is correctly recovered. When only local constraints are used and no global consistency is enforced, the resulting completion leads to inconsistencies (c). The background figure is wrongly recovered twice with wrong motion trajectories [WSI07].

segmentation that decomposes a target video into color and illumination layers. The periodicity of structured moving objects – so-called *movels* – is sampled and aligned with partially damaged movels to complete missing foreground regions. Here, a 3-dimensional tensor voting technique is used to maintain the consistency in both the spatio-temporal and the illumination domain. However, this approach is limited to objects that exhibit cyclic motions and requires an important amount of user interaction, as the user has to manually draw the boundaries of the different depth layers of the sequence. Cheung et al. [CZV06] use object templates extracted at other time instances as candidates for interpolating the missing foreground information. To ensure continuous motion of these foreground objects, a cost function based on the dissimilarity between successive templates and actual objects is minimized. However, the results are unsatisfactory if the number of postures is insufficient. To account for this problem, Ling et al. [LLL+11] utilize a posture synthesis process that combines the constituent parts of different available poses to enrich the contents of a posture database. A graphical model is constructed that predicts the motion tendency by projecting all postures onto a feature space according to a shape context descriptor. Based on this model, suitable poses are obtained to replace damaged or missing postures by finding an approximate path that links data points in this low-dimensional manifold. Finally, a MRF model is applied to compute an overall best solution, and the potential trajectory with the maximum total probability is taken as the final result. However, as in all object based algorithms, if the object segmentation is not done accurately it will lead to visually unpleasant artifacts.

Surveys of digital video inpainting techniques are presented in [APS12; AD13].

2.4 Disocclusion filling

This thesis addresses the problem of disocclusions in the field of DIBR that occurs when regions covered by foreground objects in the original view are disclosed in the synthesized virtual views. Since the filling of disocclusions can be viewed as a particular area of image and video completion [TLD07], various well-established inpainting approaches have been adopted to provide realistic content in the disoccluded regions. The most prominent example in this regard is the method of Criminisi et al., which was extended in several works by taking the available depth information into account. Daribo and Pesquet-Popescu [DP10] first introduced an additional depth term $D(\mathbf{p})$ in the priority calculation (*cf.* Equation 2.2):

$$P(\mathbf{p}) = C(\mathbf{p}) \cdot D(\mathbf{p}) \cdot L(\mathbf{p}) \quad , \quad (2.4)$$

where $L(\mathbf{p})$ is the level regularity term, which favors the inpainting of background pixels over foreground ones. $L(\mathbf{p})$ is defined as the inverse variance of the depth patch $Z_{\mathbf{p}}$:

$$L(\mathbf{p}) = \frac{|Z_{\mathbf{p}}|}{|Z_{\mathbf{p}}| + \sum_{\mathbf{q} \in Z_{\mathbf{p}} \cap \Phi} (Z_{\mathbf{p}}(\mathbf{q}) - \overline{Z_{\mathbf{p}}})^2} \quad , \quad (2.5)$$

where $|Z_{\mathbf{p}}|$ is the area (i.e. the number of pixels) of the depth patch $Z_{\mathbf{p}}$ centered at \mathbf{p} , $Z_{\mathbf{p}}(\mathbf{q})$ is the depth value at the pixel location \mathbf{q} in the source region Φ under $Z_{\mathbf{p}}$ and $\overline{Z_{\mathbf{p}}}$ the mean value

of pixels in Z_p . Similar approaches including data terms based on structure tensors and CCD have been presented in [CLL11; AK12; CLH12; MOS13].

Furthermore, Vázquez et al. [VTS06] proposed a set of fast and simple techniques for disocclusion inpainting. For instance, *constant color filling* denotes a method to complete a hole with a unique and constant color by averaging the color values at the hole boundary according to a 4-connected neighborhood. Another approach is mentioned that *horizontally interpolates* the information of the hole boundary, thus foreground and background objects are fused together to fill the gaps caused by depth discontinuities. In contrast, *horizontal extrapolation* takes the depth information into account to avoid foreground information in the filling process of the holes. Particularly, the values of pixels on the boundary that belong to objects of the local background are extrapolated to synthesize the missing information. A similar approach is revisited in [ESG12], but instead of using a single color value at each scanline, the vacant areas are completed by replicating small portions of the adjacent background regions. This method is used for evaluation in this thesis and will be elucidated in detail in Section 3.10.1.

Moreover, several video inpainting approaches have been presented in the field of stereo vision. In [KND+10] and [KWD+12], the problem of disocclusion is tackled by continuously updating a so-called background sprite, which stores the background information of previous frames and by image registration of subsequent frames to compensate global background motion between temporally neighboring frames, respectively. Schmeing and Jiang [SJ10] used background subtraction to segment each frame into background and foreground assuming a static camera setup. In order to minimize spatial and stereo discrepancies, Raimbault et al. [RPK12] reconstructed motion and inter-frame disparity vectors as guides for finding appropriate example source patches from different portions of the video sequence. However, like the majority of conventional video inpainting techniques, also stereo-vision-related methods mainly use common image completion approaches to fill the remaining holes.

Recently, stereoscopic image inpainting approaches which utilize PatchMatch-based correspondences search have been presented in [MHC+12] and [HBG11]. Morse et al. [MHC+12] extended their patch distance metric to encourage stereo consistency by penalizing patches that exhibit visual dissimilarity at the relevant disparity. In addition, they explicitly allow the matching of patches across stereo image pairs to provide a richer set of source patches. However, this kind of information does not exist in a 2D-plus-depth setup as considered in this thesis, where only a single image frame and its corresponding depth map are provided as input. A method according to the proposed framework has been presented by He et al. [HBG11], where the PatchMatch algorithm is adapted to include the additional depth information for the purpose of object removal. However, the authors use the original disparity values (before object removal) as depth constraints, which are not available in a free-viewpoint environment, as discussed in Section 3.7.

Algorithm and Implementation

This chapter gives a step-by-step overview of the developed digital inpainting approach. Starting from the initial PatchMatch algorithm of Barnes et al. [BSF+09], which constitutes the foundation of the proposed framework, the drawbacks of their patch matching method are analyzed and several adaptations and refinements thereof are presented. These adaptations comprise additional preprocessing steps, adaptive patch sizes, a matching threshold, as well as depth-based and temporal extensions. Further algorithmic refinements covering the patch distance metric, the choice of the color space and the actual pixel color synthesis are discussed in addition. Finally, the two additional inpainting approaches used for evaluation, Adobe[®]'s *content-aware fill* available since Photoshop[®] CS5 and a *horizontal background replication*, are elucidated.

3.1 PatchMatch

In 2009, Barnes et al. [BSF+09] proposed an efficient method to address the problem of finding correspondences between disjoint image regions, named *PatchMatch*. In their approach, for every $l \times l$ squared window – also referred to as a *patch* or a *fragment* – in the *target image A*, the task is to find the approximate nearest neighbor patch in the *reference image B* under a specific patch distance metric. This mapping is called a *Nearest-Neighbor Field (NNF)*, which is schematically illustrated in Figure 3.1.

In their work, a NNF is a function $f : A \mapsto \mathbb{R}^2$ of offsets, defined over all possible patch coordinates (locations of patch centers) in image *A*, for some distance function D between two patches. Given patch coordinate $\mathbf{a} = (x_a, y_a)$ in image *A* and its corresponding *Nearest Neighbor (NN)* $\mathbf{b} = (x_b, y_b)$ in image *B*, $f(\mathbf{a})$ is simply $\mathbf{b} - \mathbf{a}$. In contrast to this formulation of function f using relative coordinates (offsets), the mapping can likewise be specified in an absolute manner (*cf.* [BSG+10]). In this case, the output of $f(\mathbf{a})$ reduces to \mathbf{b} .

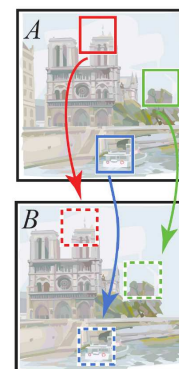


Figure 3.1:
Nearest-neighbor
field [BSF+09].

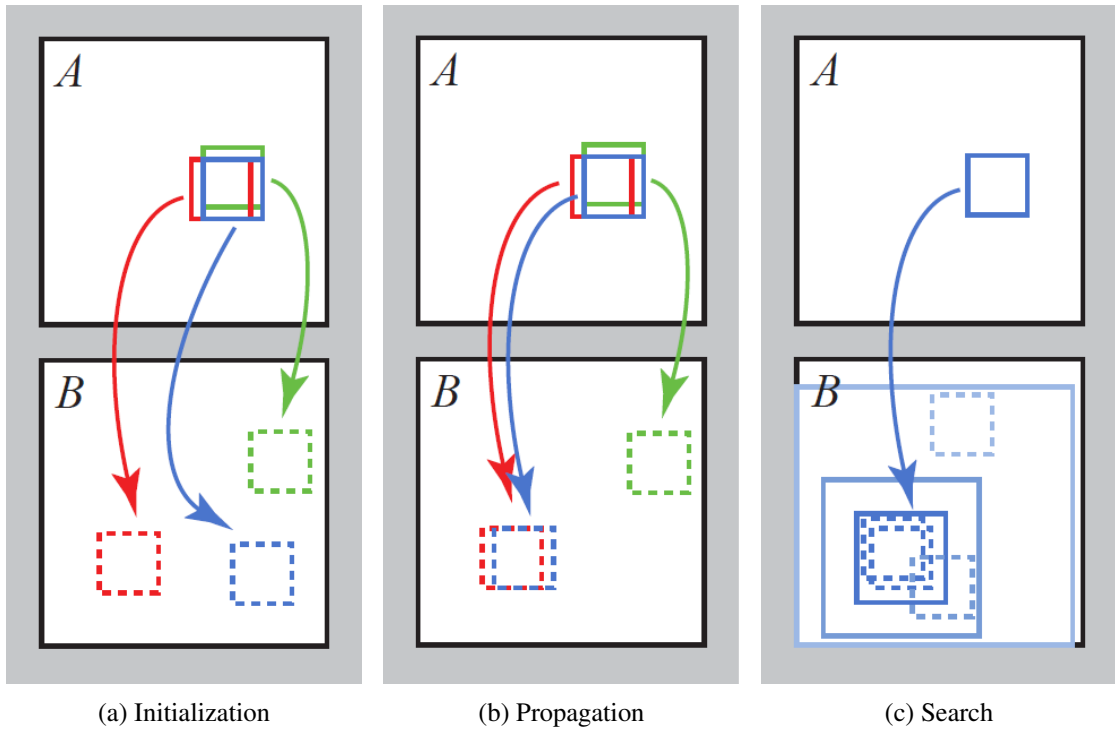


Figure 3.2: Phases of the randomized NN algorithm: (a) patches initially have random assignments; (b) the blue patch checks above (green) and left (red) neighbors to see if they will improve the blue mapping, propagating good matches; (c) the patch searches randomly for improvements in a concentric neighborhood [BSF+09].

Since approaching the problem of determining a NNF with a naïve brute force search is computationally expensive, i.e. $\mathcal{O}(mM^2)$ for image regions and patches of size M and m pixels, the randomized algorithm of Barnes et al. performs an iterative process of improving the NNF f until convergence. The key insights that motivated their approach for finding these so-called *Approximate Nearest-Neighbor Fields* (ANNF) are that adjacent offsets search cooperatively and that even a random offset is likely to be a good guess for many patches over a large image. Figure 3.2 summarizes the three main components of their algorithm:

- Initialization
- Propagation
- Search

Initially, the ANNF is either filled with random coordinates uniformly sampled across image B or by using prior information. After initialization, the algorithm works by iteratively improving the ANNF, where each iteration proceeds as follows: Offsets are examined in scan order (from left to right, top to bottom) and each offset undergoes *propagation* followed by

random search. These operations are interleaved at the patch level: if P_j and S_j denote, respectively, propagation and random search at patch j , then the processing order is given by: $P_1, S_1, P_2, S_2, \dots, P_n, S_n$.

The propagation step attempts to improve the offset $f(\mathbf{a})$ of a NN using the known offset of adjacent neighbors above or to the left (see Figure 3.2(b)). Hence, the new possible candidates for $f(\mathbf{a})$ are $f(\mathbf{a} - \mathbf{v})$, where \mathbf{v} denotes a field vector taking on the values of $(1, 0)$ and $(0, 1)$, respectively. In particular, let $D(\mathbf{v})$ denote the patch distance (error) between the patch at (x, y) in A and patch $(x, y) + \mathbf{v}$ in B . The new value for $f(x, y)$ is obtained as the *arg min* of $\{D(f(x, y)), D(f(x - 1, y)), D(f(x, y - 1))\}$. The effect is that if (x, y) has a correct mapping and is in a coherent region R , then all of R below and to the right of (x, y) will be filled with the correct mapping. Moreover, on *even* iterations propagation is done in reverse scan order, and candidates below and to the right are examined, so information propagates *up* and *left*.

Propagation converges very quickly, but if it is used alone it might end up in a local minimum. Therefore, the second step of each iteration utilizes random search by testing a sequence of candidate offsets at an exponentially decreasing distance. Let $v_0 = f(x, y)$, the candidate offset u_i is defined as follows:

$$u_i = v_0 + w\alpha^i \mathbf{R}_i, \quad (3.1)$$

where \mathbf{R}_i is a uniform random value in $[-1, 1] \times [-1, 1]$, w is a large maximum search radius, and α is a fixed ratio between search window sizes. The index i is increased from $i = 0, 1, 2, \dots, n$ until the search radius $w\alpha^i$ is below 1 pixel. Barnes et al. [BSF+09] set w to the maximum image dimension, and $\alpha = \frac{1}{2}$.

3.2 Hole filling using initial PatchMatch

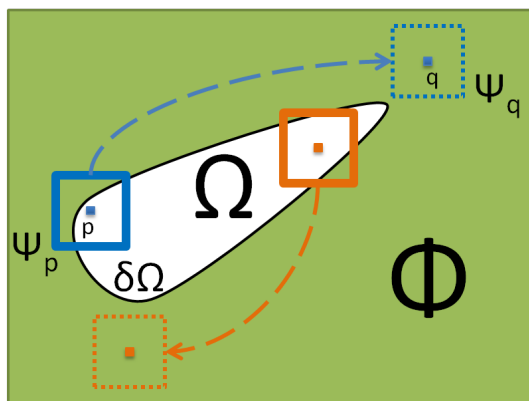


Figure 3.3: Source and target regions in the inpainting process: Ω denotes the target region (“hole”) to be filled and $\delta\Omega$ its boundary. The source region is indicated by Φ , which provides samples used in the inpainting process. For every pixel location p (shown as small squares in blue and orange) in the hole region, the associated target patch Ψ_p (square with solid lines) is mapped to a source patch Ψ_q (square with dashed lines), which contains valid visual information.

Now, using the work of Barnes et al. as the instrument for finding patch correspondences, the initially proposed PatchMatch algorithm is applied to the area of image inpainting. For the sake of comprehensibility, the formalism of the inpainting problem as briefly discussed in Section 2 is recapped, where the notation similar to that used in the inpainting literature is adopted for the ease of comparison: Let I be an input image and $\Omega \subseteq I$ a “hole” to be filled, called the *target* region. That is, Ω denotes all the missing pixels within I . Additionally, the *source* region Φ , which remains fixed throughout the algorithm, provides samples used in the filling process. Typically, $\Phi = I \setminus \Omega$, so the remaining image portions outside the hole are used to fill in the blank regions. Note that this means that the target image and the reference image coincide. The goal is now to complete the missing region Ω with some new data so that the resulting images will be visually coherent. In Figure 3.3, source and target regions as well as source and target patches are schematically illustrated.

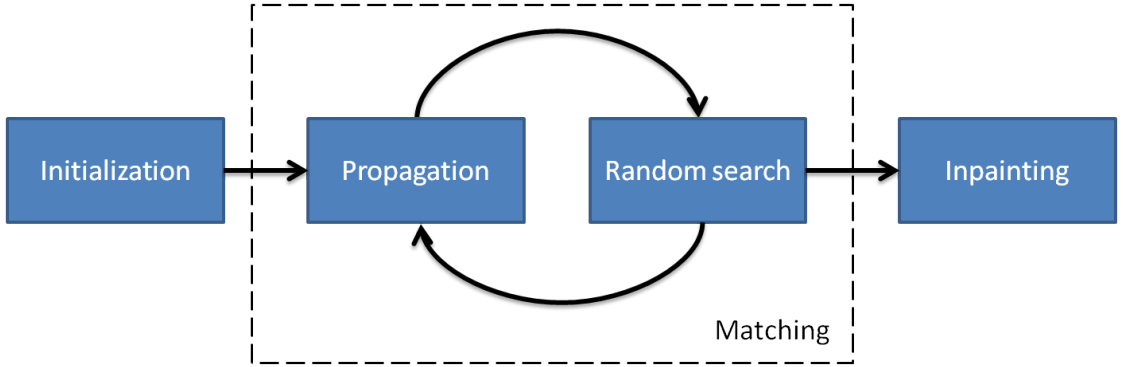


Figure 3.4: Workflow of PatchMatch-based image completion using onetime inpainting.

Figure 3.4 visualizes the overall workflow of the proposed image completion system: After the pixels in the hole areas are initialized with random NN locations, several iterations comprising propagation and random search are employed to improve these patch correspondences. Here, the *Sum of Squared Differences* (SSD) is employed as the patch distance metric:

$$D(\Psi_p, \Psi_q) = \sum_{(x,y)} \|\Psi_p(x, y) - \Psi_q(x, y)\|_2^2, \quad (3.2)$$

where (x, y) denotes every pixel location within the target patch Ψ_p and the source patch Ψ_q , respectively. Finally, the target regions of the image are actually filled by averaging the color values of overlapping patches at each pixel position. In the proposed framework the inpainting step is only performed once at the end of the image completion chain. In other words, the ANNF is solely built on already present visual information of the source region, which is equivalent to putting no confidence into newly synthesized image data. The advantages of this approach are twofold: since the actual inpainting stage is conducted only one time, the overall computation time is reduced. Additionally, the risk of leading the inpainting process in an inauspicious direction due to poorly initialized patch correspondences is avoided. However, once an empty pixel is filled up there is no chance to modify the color even when obvious conflicts among neighboring pixels appear later.



(a) Input color image

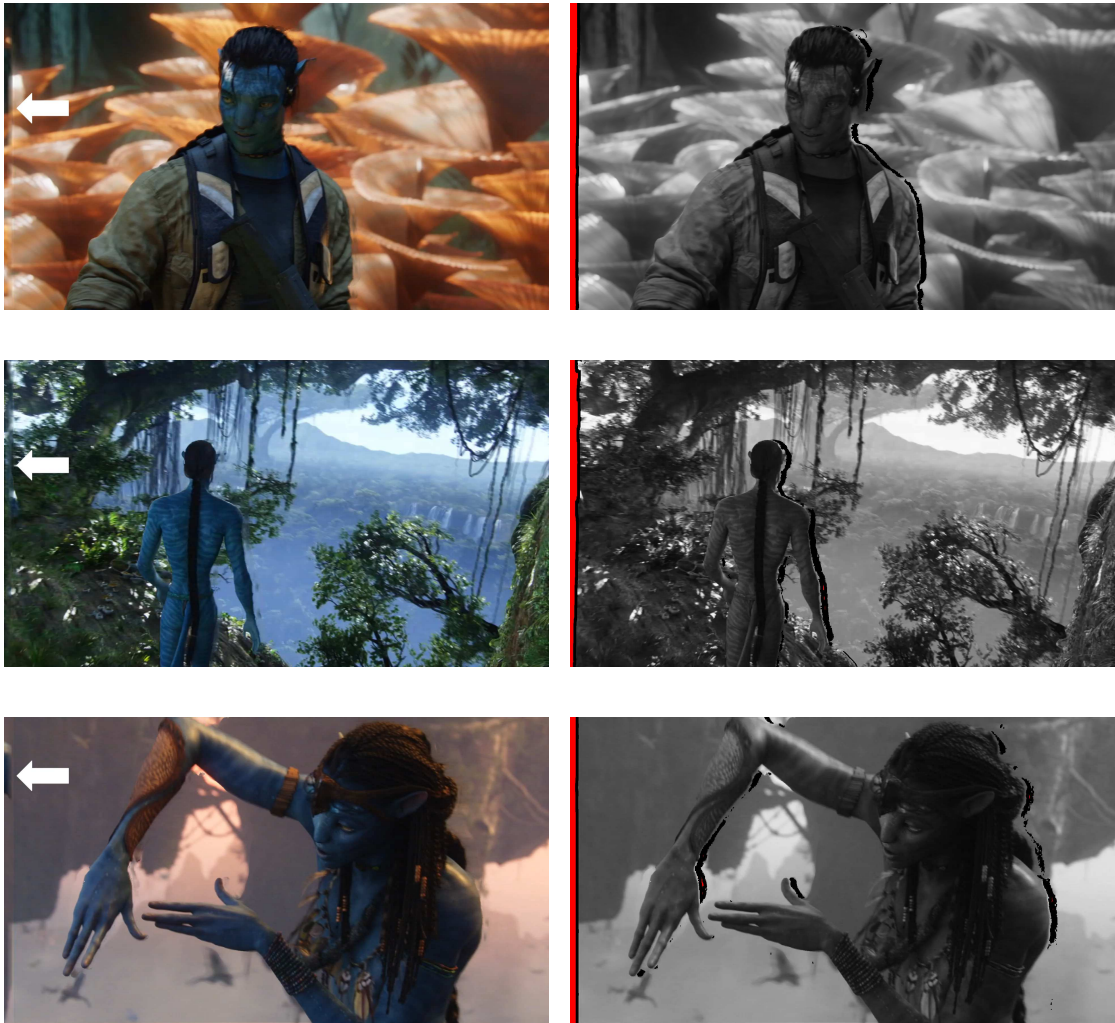
(b) Disocclusion mask

Figure 3.5: System input comprising color image and corresponding disocclusion mask. Here, black image borders are added to visualize the delimitation of the figures (see margin on the left).

The proposed inpainting system takes a 4-channel ARGB color image as input, where the alpha channel A indicates the existing holes in the scene. In Figure 3.5, three suchlike example input images are shown in conjunction with their corresponding disocclusion masks.

3.3 Non-occupied target patches (“Blanks”)

The resulting image completion outcomes using different patch sizes are presented in Figure 3.6(a) and Figure 3.7(a). As can be seen especially at the left edge of the images, there



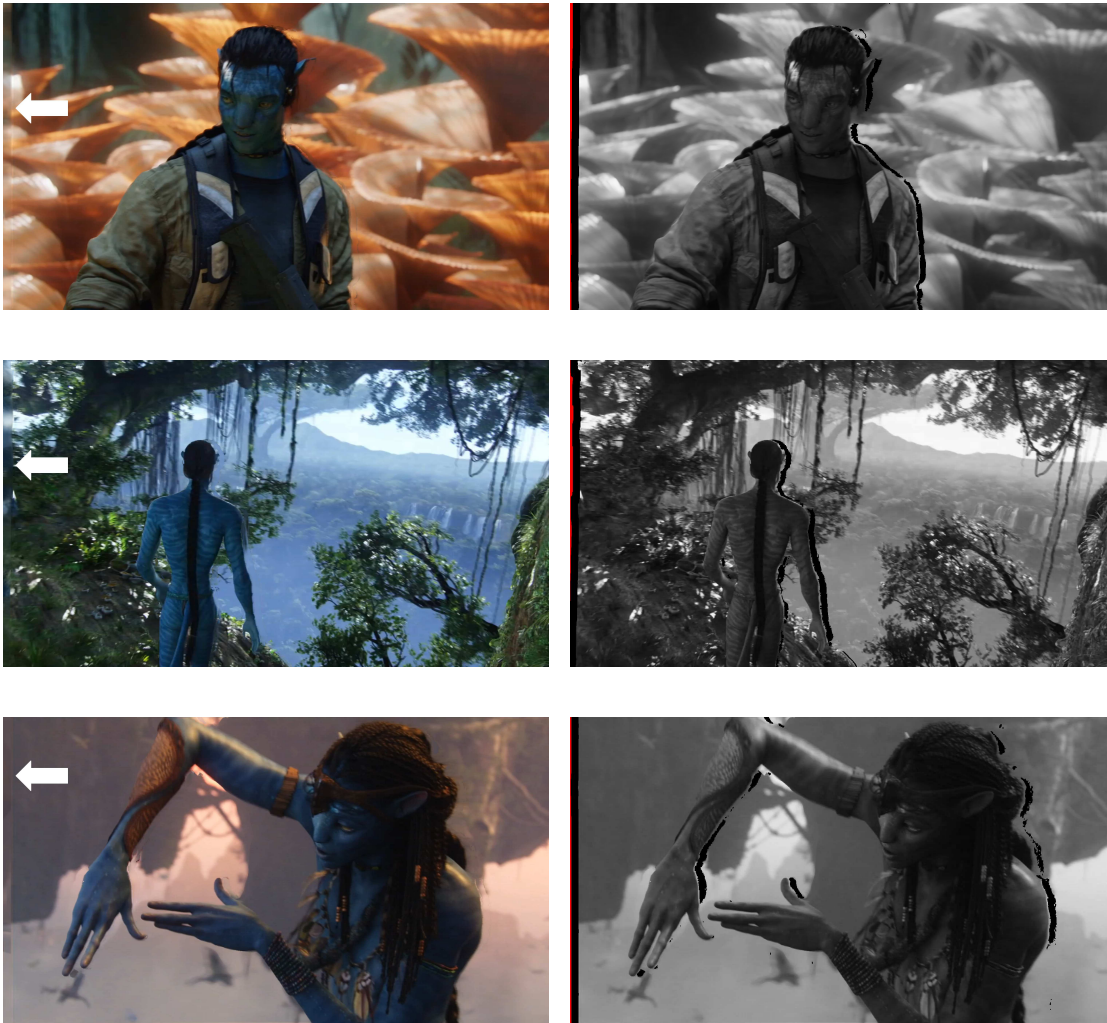
(a) Completion results

(b) Holes and blanks

Figure 3.6: Image completion examples using initial PatchMatch with fixed patch size of 21 pixels: (a) random inpainting results in the area of blank patches (note left margin of the image); (b) gray scale version of input image including holes, totally non-occupied patches are additionally marked in red.

exist a large amount of inconsistencies. These inconsistencies come from image regions where no visual information is available for the computation of the cost function, which is based on the color similarities of matching patches. In particular, when the patch size l is smaller than the size of Ω , there must be some totally blank patches containing no valid pixels. In Figure 3.6(b) and Figure 3.7(b), these image regions are marked in red and are subsequently termed as “blanks”.

By comparing these two figures, it can be seen that as the patch size increases the number of blanks reduces, which produces a coherently improved visual impression. However, since



(a) Completion results

(b) Holes and blanks

Figure 3.7: Image completion examples using initial PatchMatch with fixed patch size of 51 pixel: (a) the amount of non-occupied patches and consequently randomness are reduced, but blurrier inpainting results due to bigger patch sizes; (b) gray scale version of input image including holes, totally non-occupied patches are additionally marked in red.

the range of influence increases for larger patches, each patch contributes to a higher number of pixels in its vicinity. Consequently, the inpainting result becomes much blurrier, if overlapping patches do not agree on the color value of an individual pixel.

Now, let us take a closer look at the scenarios that arise when considering the different filling degrees of source and target patches, as listed in Table 3.1. Since target patches are placed at hole regions where missing information is supposed to be filled in, they can obviously not be completely filled. Likewise, source patches provide valid samples in the course of the inpainting

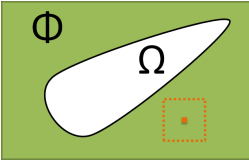
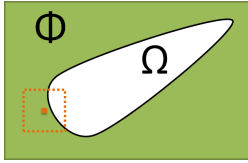
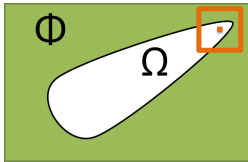
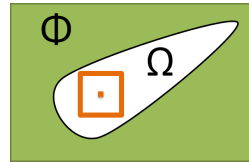
<i>Information levels</i>	Full	Partial	Empty
<i>Patches</i>			
Source			
Target			

Table 3.1: Potential levels of information for source and target patches

process, therefore these kinds of patches cannot be vacant. Additionally, as mentioned before, target patches may be completely empty (blanks) or – as in most cases – partially occupied. Accordingly, source patches are entirely filled in the best case or only filled in parts.

With these scenarios in mind, the requirements for a workable image completion pipeline can be stated as follows:

1. Matching requires a minimum number of valid corresponding pixels.
2. Inpainting requires valid pixels from source patches.

Furthermore, as the binary masks indicating valid pixels of a target and source patch are combined using the logical AND operator, different overlapping scenarios of corresponding patches occur in the matching step. As seen in the areas marked in red in Figure 3.6 and Figure 3.7, if totally blank patches are present in the target region then no reasonable patch correspondences can be calculated due to zero overlap. Moreover, there may arise situations where the combination of partial target as well as partial source patches also violates the first requirement (see Figure 3.8). Hence, the only case where the combined mask is guaranteed not to be empty, is when all entries in the source patch are valid.

To recap, whilst blanks entail a crucial drawback when considering the first requirement, partial source patches cause problems especially in the context of the second one. Additionally, it should be noted that the sets of pixels used for matching and inpainting are disjoint. In other words, the set of valid pixels in a sample is composed of the matching and the infilling portion.

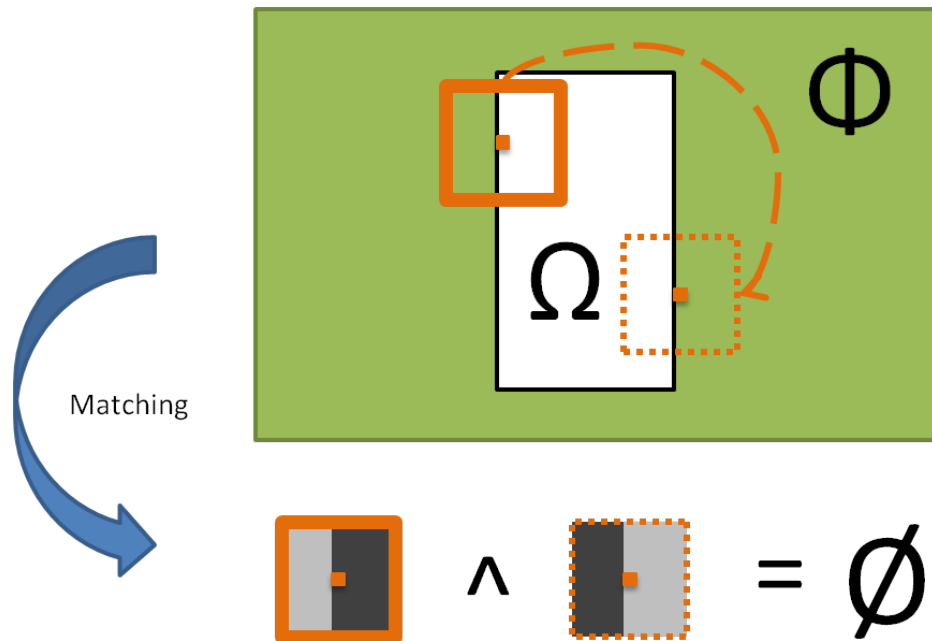


Figure 3.8: Example of empty overlapping set caused by a partial source patch. Valid pixels are marked in light gray and invalid ones are colored in dark gray. When the patches are combined to calculate their similarity during the matching step, no information is left due to non-overlapping positions of valid pixels.

3.4 Additional preprocessing steps

When we investigate the inpainting results in Figure 3.6 and Figure 3.7 more precisely, two other types of problems are perceived that degrade the overall performance of the image completion process: warping artifacts and “small” holes. These issues will be elucidated in detail in the following subsections.

3.4.1 Single-sided mask dilation

The example images shown in Figure 3.5 are generated by warping the left image onto the position of the right one. Consequently, artifacts caused by inaccuracies in the underlying disparity maps are present especially at the right side of a hole region. An example of these warping artifacts are illustrated in Figure 3.9. As can be seen, parts of the foreground object are torn apart at the hole periphery, here in particular the right ear of the character.

Since the determination of the ANNF is based on the color difference of corresponding patches, these artifacts significantly impair the quality of the matching and inpainting results. Hence, as an additional preprocessing step, several iterations¹ of morphological dilation are

¹Note that in this case the dilation can also be employed only one time by using a respective large mask.

applied to the binary disocclusion mask using the following structure element E . In our system, the number of iterations was empirically set to 7.

$$\begin{array}{ccc}
 E = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} & | & E = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \\
 \text{if warped to the left} & & \text{if warped to the right}
 \end{array}$$

Figure 3.9(b) shows that after the usage of unilateral morphological dilation the aforementioned artifacts have nearly vanished. However on the downside, valid visual information gets lost as well. Moreover, due to the growth of the hole regions, the number of pixels to be filled in-



(a) Input image snippet



(b) After dilation

Figure 3.9: Single-sided mask dilation as preprocessing step to reduce warping artifacts, where parts of the foreground object (here: right ear) are torn apart at the hole periphery due to inaccuracies in the underlying disparity map.

creases, which implies additional computation cost. Clearly, the amount of totally blank patches rises as well.

3.4.2 Diffusion-based infilling of “small” holes

The quality of a resultant novel view crucially depends on the pixel accuracy of the underlying disparity map, which constitutes a fundamental component during the warping stage. Thus, minor inaccuracies in the disparity map may cause image regions to be spuriously labeled as disocclusion, although these pixels together with their immediate neighbors are located on the same depth level in the original view. Typically, these regions contain only a small number of pixels compared to the full image resolution. The problem of filling such “small” holes is now addressed as a second preprocessing step.

Since these spurious disoccluded regions are of small sizes, simpler models can be used to locally approximate the inpainting results produced by more sophisticated ones. Consequently, the diffusion-based algorithm of Oliveira et al. [OBM+01] is applied which is designed on the observation that the human visual system can tolerate some amount of blurring in areas not associated to high contrast edges [K79]. In particular, let Ω be a small area to be inpainted and let $\delta\Omega$ be its boundary. Since Ω is small, the inpainting procedure can be approximated by an isotropic diffusion process that propagates information from $\delta\Omega$ into Ω . It should be noted that convolving an image with a Gaussian kernel (i.e. computing weighted averages of pixels neighborhoods) is equivalent to isotropic diffusion characterized by the linear heat equation. The simplest version of the algorithm consists of initializing Ω by clearing its color information and repeatedly convolving the region to be inpainted with a diffusion kernel. Table 3.2 shows two such kernels that only consider contribution from the neighbor pixels, i.e. it has a zero weight at the center of the kernel [OBM+01].

a	b	a
b	0	b
a	b	a

c	c	c
c	0	c
c	c	c

Table 3.2: Two diffusion kernels for isotropic inpainting.

$$a = 0.073235, b = 0.176765, c = 0.125.$$

In Figure 3.10(a), a section is outlined where disparity map inaccuracies occur with high frequency. In our approach, the resulting holes are filled in the color image as well as the corresponding disocclusion mask (see Figure 3.10(b,c)). This implies that the synthesized visual information introduced at the preprocessing stage becomes valid throughout the subsequent matching and inpainting steps. The advantage of filling these little gaps as early as possible in the image completion workflow are twofold: since at every matching iteration valid image data are omitted in the construction of an ANNF, the overall processing time is reduced. Additionally, as the patch size increases, the ratio between matching and inpainting pixels becomes imbalanced, which leads to erroneous infilling results. As can be seen in Figure 3.10(e), these corrupted regions are avoided in contrast to Figure 3.10(d), where the PatchMatch algorithm is applied using a patch size of 51 pixels. In this thesis, a hole is defined to be “small”, if its area



(a) Input color image with marked magnification area



(b) Input disocclusion mask



(c) After small hole filling



(d) Inpainting result using PatchMatch



(e) Inpainting result using isotropic diffusion

Figure 3.10: Inpainting of small holes using isotropic diffusion.

occupies less than 0.5 per mill of the image resolution. Individual hole areas are determined by *connected component labeling* [RP66].

3.5 Adaptive patch sizes

In most of the existing exemplar-based inpainting algorithms, the patch size is fixed as a default number or specified by the user [ZZ10]. However, the patch size has an important influence on the image completion quality because it affects how well the filled patch captures the local characteristics of the source image. In this thesis, adaptive patch sizes are introduced to additionally address the problem of blanks as can be seen in Figure 3.11.

For that purpose, a threshold is specified to ensure a minimal amount of valid pixels in each target patch, which allows to calculate the respective similarity between source and target fragments. The corresponding patch size for each target pixel is determined by constantly incrementing the patch dimensions until the percentage of source pixels exceeds a predefined value. Hence, patches become smaller near the hole border and are continuously growing towards the hole's centroid. This approach also helps to gently propagate information from target border regions, since the overlap of neighboring patches close to the hole boundary is reduced and consequently fewer patches are involved in the calculation of the color value of an individual pixel. This strategy is additionally justified by the fact that for patches situated at the periphery of a hole, the valid visual information is located closer to the patch center, which represents the actual pixel being inpainted. In contrast, for patches that are centered at the interior of the hole, the valid pixels lie on the exterior of the patch (see Figure 3.11). Therefore, since the source pixels are located more remotely compared to the boundary patches, the information of a greater number of overlapping patches is used to calculate the color value of an internal hole pixel.

Depending on the input data, adaptive patch sizes may also reduce the overall computation time, since the number of individual pixel comparisons decreases for smaller patches, which denotes the most time consuming processing step in the image completion pipeline. Additionally, note that the patch dimensions have to be clamped to the image bounds when determining the ratio of valid pixels. For example, a squared patch located at the top-left corner of an image has solely an effective size of one quarter of the original size. Thus, the patch magnitudes have to

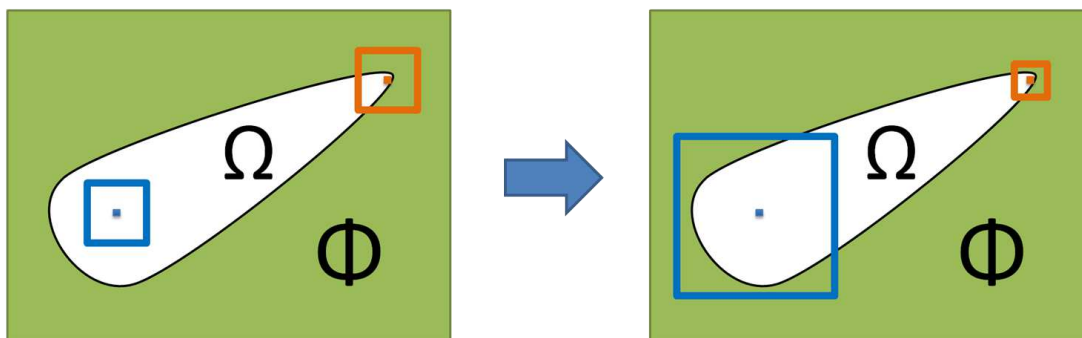


Figure 3.11: Adaptive patch sizes are used to address the problem of blanks.

be adapted simultaneously as the fragments grow to facilitate the convergence of the patch size determination.

In this work, the threshold of valid pixels in a target patch is defined to be 10 percent of the particular patch size. The minimal and maximal patch size values are set to 5 and 161, respectively. This upper bound is introduced to prevent the fragment size from becoming too large, if the specified percentage of valid pixels cannot be achieved. Although a patch size of 161×161 already appears to be enormous, such extents mainly occur at the image borders, where the dimensions get cropped anyway as mentioned before. Moreover, integral images [VJ01] are used for faster computation of the patch sizes.

3.6 Valid matching pixel threshold

By introducing adaptive patch sizes it is guaranteed that the majority of the target fragments contain a certain percentage of valid pixels which are utilized in the matching step. However, as mentioned in section 3.3 and schematically illustrated in Figure 3.8, there may arise situations where the combination of target and source patches becomes impractical. Such an example is presented in Figure 3.12 and subsequently discussed in detail.

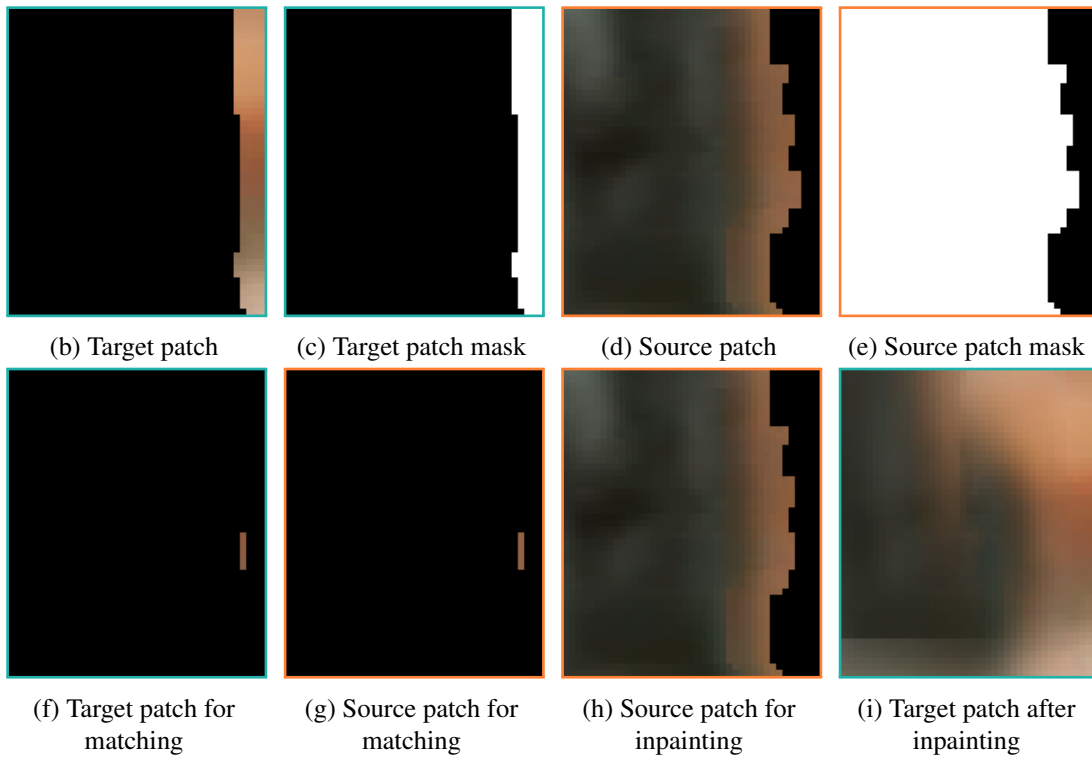
Figure 3.12(a) shows a sample frame where the inpainted regions (holes) are outlined in red. One can clearly see the visual disturbances on the left side of the image. Let us take a closer look at one target patch (marked by a cyan rectangle, see also Figure 3.12(b)) and its corresponding source patch (marked by an orange rectangle, see also Figure 3.12(d)) in this area before the inpainting process has taken place. Note that the initial squared target patch – and as a consequence also the source fragment – is cropped at the left image border. Their corresponding masks are shown in Figure 3.12(c) and (d). These binary masks indicate the amount of valid (i.e. non-hole) pixels which are used during the matching and inpainting phase. Since the target patch is located at a hole region on the left edge of the image, source pixels are appended on the right side of the patch as the fragment size has grown. On the other hand, the corresponding source patch is situated in an area where a hole appears on the right side of the patch.

Since a pixelwise cost function can only be determined if a pixel is valid in the same position in both target and source patch, the mask of both fragments have to be logically combined to calculate the similarity between the two patches. Figure 3.12(f) and (g) show the remaining portion of matching pixels in the target and source patch, respectively. In particular, only a 1×5 pixel strip is left where the color values coincide in both fragments. Thus, the respective difference between these patches is small, which is equivalent to a high rated similarity. Finally, the remaining (valid) part of the source patch (see Figure 3.12(h)) is used to fill in the vacant pixels in the target patch which are indicated by the inverse target mask. Since the target fragment protrudes an object edge, different color values are present in the left and right half of the patch, which results in the deficient inpainting outcome.

Apparently, the imbalance between matching and inpainting pixels constitutes the primary cause of the unsatisfactory image completion performance. In particular, in the presented example only 0.3% of the valid pixels are matched whereas 99,7% are used for inpainting. Hence, a threshold is specified to ensure a minimal amount of valid pixels in the matching step. In other words, since a target patch contains only a small number (e.g. 10%) of valid pixels, the aim



(a) Sample frame including target (cyan) and source (orange) patch. Holes are marked in red.



(b) Target patch

(c) Target patch mask

(d) Source patch

(e) Source patch mask

(f) Target patch for matching

(g) Source patch for matching

(h) Source patch for inpainting

(i) Target patch after inpainting

Figure 3.12: Visual disturbance caused by an imbalanced ratio of matching and inpainting pixels.



(a) Inpainting result without matching threshold



(b) Inpainting result with matching threshold

Figure 3.13: Image completion results showing the significance of a matching threshold.

is to maintain the majority of these pixels in the matching process by preferring corresponding source patches to be completely filled instead of just partially. In Figure 3.13 the image completion results with and without a valid matching pixel threshold are demonstrated, where in the

former a substantial reduction of infilling artifacts at the left image border is noticeable. As for adaptive patch sizes, the threshold is set to 10% in this work. However, recall that a source patch consists of two disjoint sets of pixels: those used for matching and those used for infilling, or briefly worded: “*what you see is not what you get*”! This demonstrates a significant drawback of the proposed approach of putting no confidence into newly synthesized image data, as stated in Section 3.2.

3.7 Depth extension

In Figure 3.13 another inpainting problem becomes apparent: *ghost shadowing artifacts*. When we take a closer look at the regions filled in next to the left arm of the character, one can see some kind of “shadow”. Since one reason for holes induced by disocclusions are sharp depth transitions at object boundaries, a target patch may comprise pixels that belong to foreground objects as well as pixels that are part of the background. As these pixels are then employed in the matching step to find a corresponding source patch, the NNS is led to an unfavorable direction. Consequently, these shadowing artifacts – hereinafter also referred to as *foreground blur* – are caused by color interference of foreground objects as source patches are preferred that have color values similar to pixels in the foreground which are afterwards utilized in the inpainting stage. Figure 3.14(a) shows a magnified version of the aforementioned area together with two additional examples.

However, since disocclusions indicate areas in an image that are covered by foreground objects in the original scene but have become visible in the newly synthesized view as the (virtual) camera had moved, it is reasonable to fill these resultant vacant regions with data obtained from the background. For this purpose, the PatchMatch algorithm is extended to incorporate depth information in the matching stage to find appropriate patch correspondences. In Figure 3.14(b) the improved results of this depth-based inpainting approach are presented, which will be precisely elucidated in the following section.

Beside the color image, the proposed image completion system now takes the disparity map as an additional input. However, to apply depth information to pixels in the target regions, holes in the disparity map have to be filled primarily. He et al. [HBG11] proposed to determine the disparity $d_{i,j}$ for each disoccluded pixel $p_{i,j}$ from its eight connected neighbors, where (i, j) denotes the pixel position. Since the missing disparity is assumed to be propagated from the adjacent background, the value of $p_{i,j}$ is selected as the $\min \{d_{i,j}\}$ of its eight candidate disparities.² The depth information in the target region is then used as a hard constraint in the exemplar filtering to reduce the candidate scope. In particular, since the disparity d_p of a blank pixel should be coherent with the disparities of its surrounding background, He et al. defined the valid disparity set $\{d_{\Psi_p}\}$ of a target patch Ψ_p as

$$\{d_{\Psi_p}\} = \{d_k \leq d_p \mid k \in \Psi_p \cap k \in \Phi\} \quad . \quad (3.3)$$

If $\{d_{\Psi_p}\} \neq \emptyset$, their second depth constraint is formulated as

$$\min \{d_{\Psi_p}\} \leq d_s \leq \max \{d_{\Psi_p}\} \quad , \quad (3.4)$$

²Note that the greater the disparity, the closer the object is located to the camera.



(a) Ghost shadowing artifacts



(b) Reduction of artifacts by depth extension of PatchMatch

Figure 3.14: Ghost shadowing artifacts caused by color interference of foreground objects.

where s of disparity d_s denotes one candidate exemplar in the source region Φ . In other words, the depth range of potential source pixels is determined by the disparity of the respective target pixel as well as the minimum and maximum disparities of the corresponding target patch. However, this approach is not applicable to the field of disocclusion inpainting due to several reasons.

Since the method of He et al. was initially proposed for foreground object removal, the value of d_p referred to the original disparity at a pixel p before the object was removed. Thus, this a priori depth information is used to ensure that foreground pixels are neglected, if a target patch overlaps both foreground and background areas. As this depth information is not available in the target regions when considering disocclusions, the disparities in the holes have to be synthesized as well. Now, since these synthesized disparities are first selected as the minimum of their vicinity and possible candidates are additionally constrained to be smaller than or equal to these values, the valid depth range is reduced to a small number of depth layers. Moreover, as disparities are propagated in all directions (8-connected neighborhood), the minimum value that borders a target region is filled into the entire hole, which finally restricts the valid exemplars to a single depth layer. This can be seen in the disparity map shown in Figure 3.15(b), where each target region (outlined in red) is filled with a single low (i.e. dark) value. Figure 3.15(c,d)

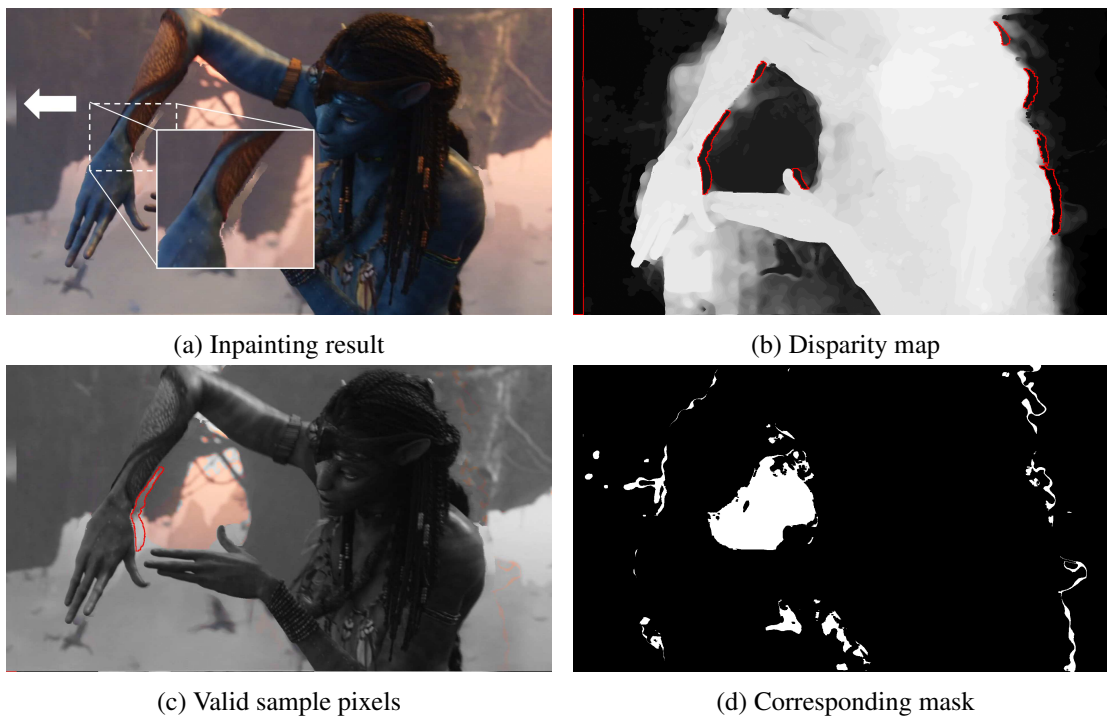


Figure 3.15: Depth-based image completion result according to He et al. [HBG11]: (a) Imperfect output due to strict depth constraints; (b) disparity map including synthesized values in the target regions (marked in red); (c) valid source pixels for one hole (marked in red), where excluded areas are grayed out and (d) corresponding binary mask of valid source pixels.

presents the valid source pixels for the hole marked in red. Due to the small number of valid candidate exemplars, patch sizes grow disproportionately to exceed both the thresholds for adaptive patch sizes, which ensures a minimal amount of valid pixels in each target patch, and the matching pixel threshold, which guarantees a minimal number of valid pixels in the matching step. Consequently, a considerable number of inconsistencies is present in the image completion result as can be seen in Figure 3.15(a).

Now, as the usage of an 8-connected neighborhood for disparity propagation would be only applicable to holes that comprise just a small amount of pixels (e.g. less than 0.5 per mil of the image resolution as defined earlier), the approach of He et al. is adapted to the needs of disocclusion infilling, where large vacant areas are present. For this purpose, depth information

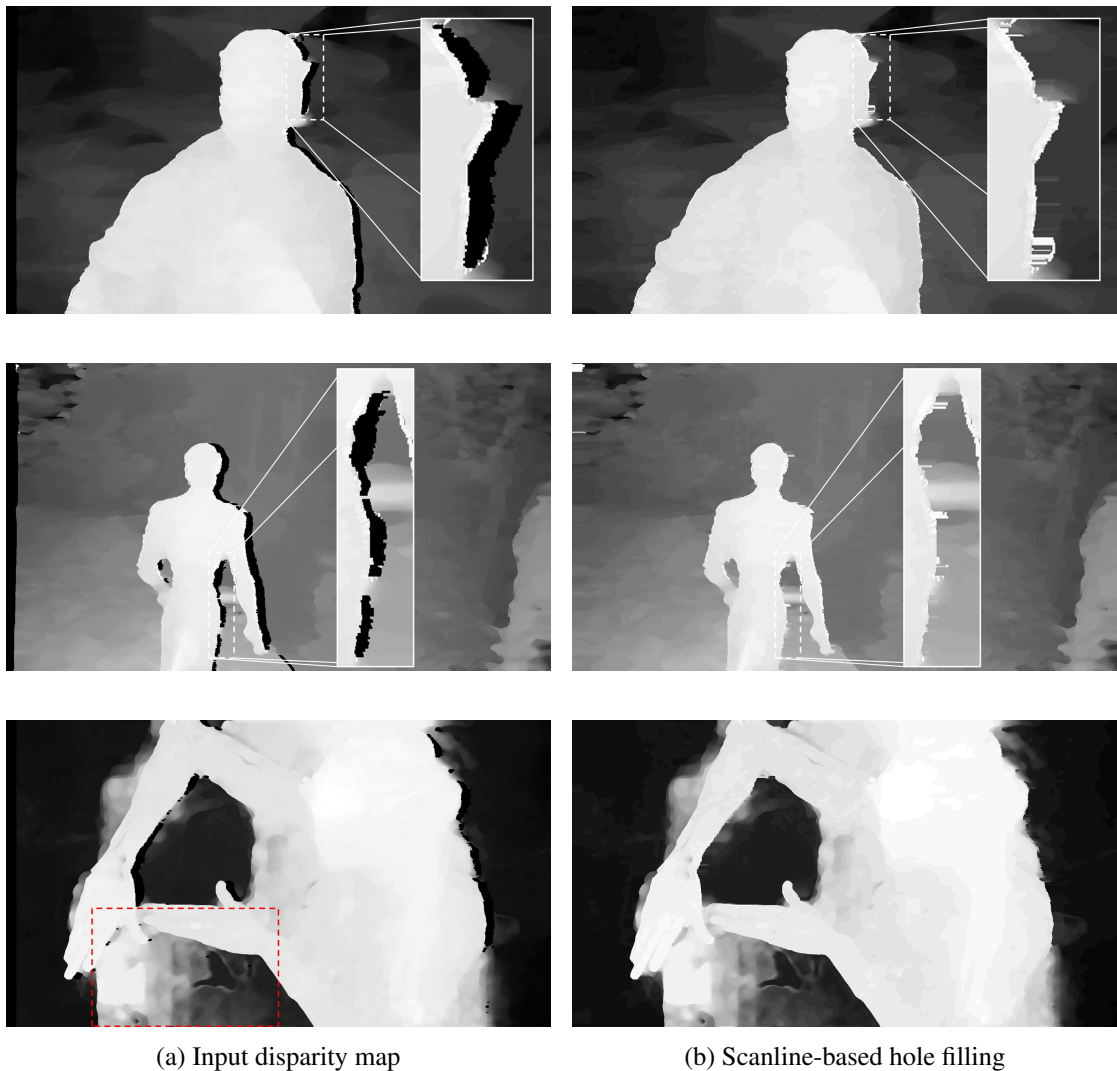


Figure 3.16: Scanline-based infilling of disocclusions in disparity maps. Warping artifacts are shown in the magnified areas and the red rectangle indicates erroneous disparities caused by the stereo matching process in untextured regions (see text for detailed description).

is restricted to be propagated solely in horizontal direction. In Figure 3.16, three examples are shown where holes are filled in a scanline-based manner by disparities of their surrounding background. Moreover, an essential problem becomes obvious: inaccuracies of the disparity maps. Many results of stereoscopic inpainting methods [VTS06; WJY+08; HBG11; MHC+12] are evaluated on ground truth data of the Middlebury dataset [SP07], which provides high-accuracy stereo depth maps obtained through a structured light technique [SS03]. However, such a technique is not applicable in the field of 3DTV and at the same time its precision and quality is not yet achieved by existing stereo matching algorithms [SS02].

As can be seen in Figure 3.16, the area marked by a red rectangle comprises erroneous depth information because the matching process for calculating stereo correspondences fails due to untextured background regions. Here, the smoothness assumption, which states that a disparity map typically consists of regions of constant or very similar disparity, is additionally violated. Consequently, to provide a reasonable number of valid candidate exemplars in the matching stage and to cope with slanted surfaces, the depth constraints of Equation 3.3 and 3.4 are alleviated. Instead of determining restrictions on the patch level, depth limits are specified for an entire hole. Thus, for every hole \mathcal{H} of the target region Ω the valid disparity set $\{d_{\Omega_{\mathcal{H}}}\}$ is defined as

$$\{d_{\Omega_{\mathcal{H}}}\} = \{d_p \mid p \in \mathcal{H} \wedge \mathcal{H} \subset \Omega\} \quad (3.5)$$

and the depth constraint is formulated as

$$0 \leq d_s \leq \max \{d_{\Omega_{\mathcal{H}}}\} \quad . \quad (3.6)$$

In other words, for each coherent target region, the maximum of the newly synthesized minimal disparity values is selected as an upper bound in the NNS whilst a lower bound is completely neglected.

However, warping artifacts (*cf.* Section 3.4.1) represent another problem as shown in the magnified areas in Figure 3.16. Since parts of the foreground object are torn apart at the hole periphery, erroneous high-valued foreground disparities are present in the background, which negatively affects the depth constraints when selecting the maximum disparity previously. Thus, a procedure is specified to exclude maximum disparity values that deviate significantly from the disparity distribution of the respective hole, as listed subsequently:

Listing Max-per-hole outlier removal

```

1: for all  $\mathcal{H} \in \Omega$  do
2:    $\mathcal{H}^* \leftarrow \mathcal{H}$ 
3:    $d_{max_{\mathcal{H}^*}} \leftarrow \max \{d_{\Omega_{\mathcal{H}^*}}\}$ 
4:   while  $\mu(\{d_{\Omega_{\mathcal{H}^*}}\}) + \alpha \cdot \sigma(\{d_{\Omega_{\mathcal{H}^*}}\}) < d_{max_{\mathcal{H}^*}} - \epsilon$  do
5:      $\mathcal{H}^* \leftarrow \mathcal{H}^* \setminus d_{max_{\mathcal{H}^*}}$ 
6:      $d_{max_{\mathcal{H}^*}} \leftarrow \max \{d_{\Omega_{\mathcal{H}^*}}\}$ 
7:   end while
8: end for

```

Here, $d_{max_{\mathcal{H}^*}}$ denotes the resulting maximum disparity value of hole \mathcal{H} , α is a constant scaling factor, ϵ declares a fixed confidence term, and $\mu(x)$ and $\sigma(x)$ denote the mean value and the

standard deviation of the disparity distribution in \mathcal{H} . In this thesis, α is set to 1.5 and ϵ is selected as 5 percent of the available depth levels in the entire image.

The enhanced image completion result according to the proposed scanline-based approach including disparity maximum selection is shown in Figure 3.17(a). In contrast to Figure 3.15(b), this maximum selection manifests itself in the higher (i.e. brighter) values in Figure 3.17(b), which are filled in the target regions outlined in red.³ Additionally, note the different amount of valid source pixels for the hole marked in Figure 3.17(c,d) and Figure 3.15(c,d). Now, only pixels of the foreground object are excluded, whereas large areas of the background remain.

Furthermore, recall the two main reasons for disocclusions that cause blank areas in a newly synthesized view:

1. Areas that are occluded by a foreground object in the original view (depth holes)
2. Areas that are outside the visible field of the original view (border holes)

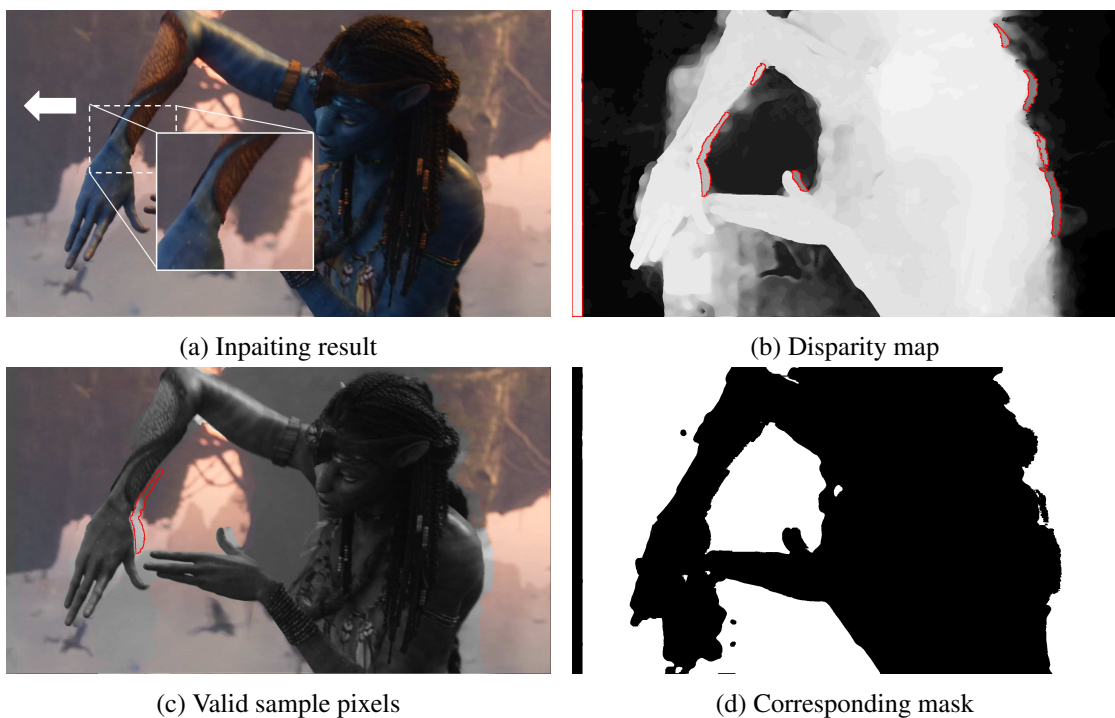


Figure 3.17: Depth-based image completion result using proposed constraints: (a) Enhanced inpainting outcome comprising a significant reduction of artifacts; (b) disparity map including synthesized values in the target regions (marked in red) based on scanline infilling and maximum selection; (c) valid source pixels for one hole (marked in red), where excluded areas are grayed out and (d) corresponding binary mask of valid source pixels.

³Note that the disparities in the target regions only indicate upper bounds for the subsequent NNS and do not express actual depth information of the corresponding synthesized pixels in the resulting image.

The proposed depth constraints account for the first cause by restricting the candidate search space to background regions. However, since holes at the image border indicate completely new areas that have become visible in the altered field of view of the (virtual) camera, it is impracticable to predict feasible depth information. On the one hand, the problem of ghost shadowing artifacts plays a minor part in areas near the image boundaries. Additionally, these border holes typically comprise large vacant regions which extend over the entire image height and cover a wide range of depth levels in the worst case. Hence, as a hard depth constraint entails a reduction of the available source pixels which results in larger target patches and consequently in a higher computation time as the number of pixelwise comparisons increases, no depth-related restrictions are used for border holes to provide a reasonable amount of candidate exemplars. This can be seen as the bright strip at the left image border in 3.17(b) which represents the highest possible depth level in the input image.

3.8 Temporal extension

The proposed image completion approach is now extended to video sequences. A naïve strategy to process video data would be to treat a video as a collection of images and perform the same computational steps on each frame independently. However, there are two main disadvantages of this approach. Due to the temporal correlation among the frames, a matching patch is also likely to be found in temporally adjacent frames, especially if a portion of the region to be filled in one frame is present in some other neighboring frame. The other disadvantage is that even if every filled frame is spatially coherent, temporal coherence is not ensured, which might result in visible artifacts in the video [KBB+05].

Consequently, a video processing algorithm that only works framewise and neglects the inter-frame relations is always in danger to produce inconsistencies that can be annoying when watching the video [SJ11]. These inconsistencies, also termed as *flickering*, correspond to rapid changes in color or brightness of pixels in successive frames.

Based on the assumption that the content of two consecutive frames will typically not differ significantly, Herling and Broll [HB10] proposed to use information of the previous frame as initialization for the subsequent one. In particular, once the image completion algorithm successfully finished the computation for the first frame, the approximated patch correspondences are used as initialization for the next frame instead of initializing the ANNF randomly. This workflow is schematically illustrated in Figure 3.18.

However, as their method was initially employed for object removal in a real-time diminished reality environment, results have shown that the benefit in the field of disocclusion filling is limited. Particularly, though the frame-to-frame propagation saves computation time as the convergence of the matching process is accelerated, the reduction of patch flickering is negligible. First, especially in dynamic scenes, where the position of the foreground objects – and accordingly the position of the disoccluded regions – alters significantly between consecutive frames, a great number of initial patch correspondences have to be discarded due to invalid image coordinates or violated depth constraints. Moreover, the scope of the NNS is still restricted to the actual frame itself.

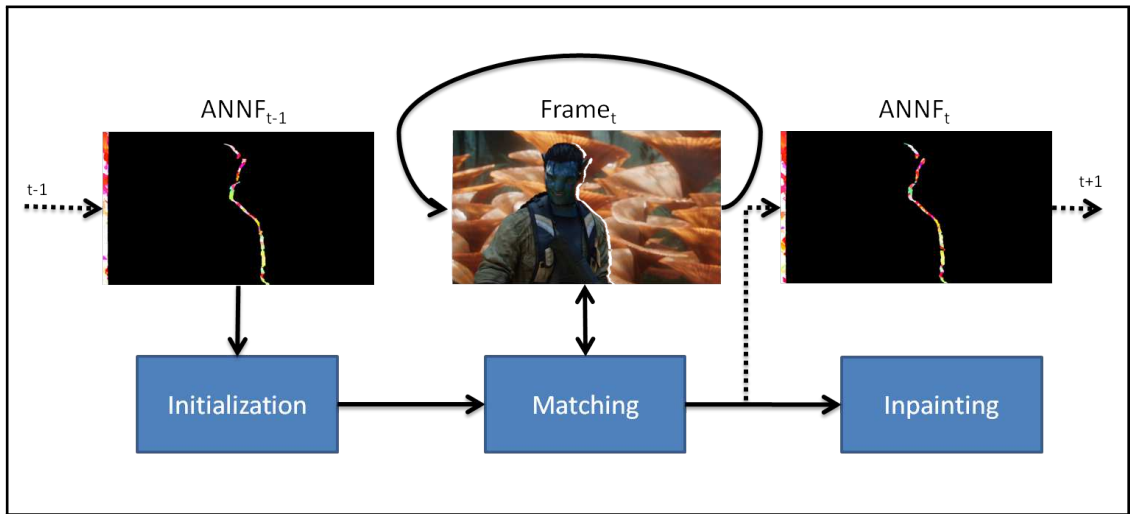


Figure 3.18: 'ANNF propagation' workflow.

When the inpainting of disocclusions is considered, also the influence of warping artifacts has to be taken into account. Since the edges of objects in an image are usually some pixels thick, they rather consist of a transition of color values than of sharp color changes due to aliasing effects [SJ11]. Now, as disparity maps define a yes-or-no decision whether a pixel belongs to the foreground or to the background, some pixels at the periphery of an object that partially feature foreground color are regarded as background. After the warping step has taken place, these erroneous pixels are present in the vicinity of the hole boundary (*cf.* Figure 3.9) where the matching pixels of a target patch are located. Thus, even slight errors in the disparity maps may

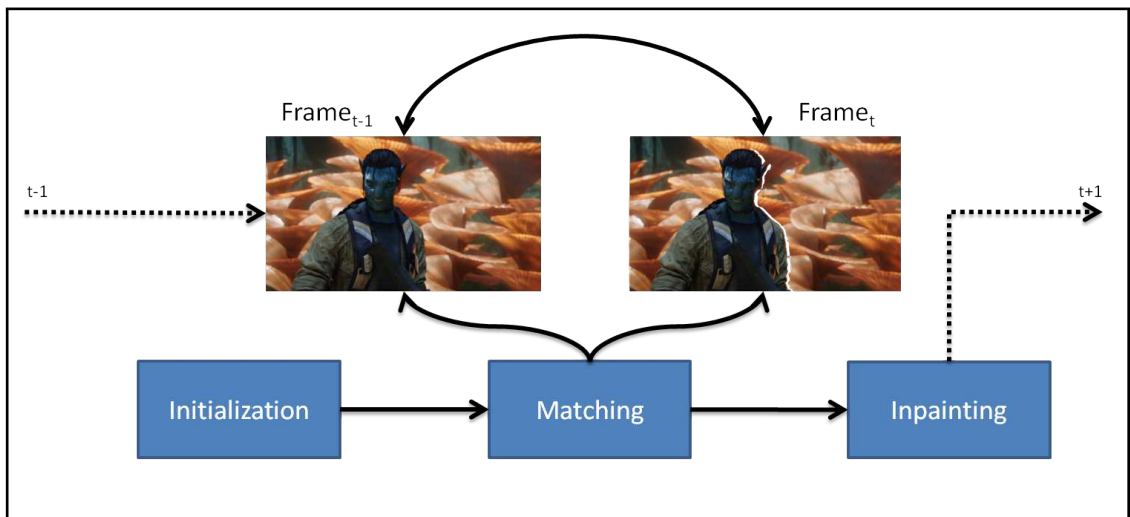


Figure 3.19: 'Previous frame as reference' workflow.

cause flickering artifacts as the edge of an object might be cut at varying positions in consecutive frames, which leads to different colors to be matched and inpainted in the target regions.

To account for the aforementioned problem, a second temporal approach is investigated. Here, the image completion result of the previous frame is taken as the reference image for the subsequent frame. This workflow is summarized in Figure 3.19. Hence, as patch correspondences are established between target patches of the actual frame and source patches of the previous one, color values of the former filled image are inherited in the inpainting process of the next frame. This entails a reduction of flickering artifacts as rapid color changes of consecutive frames are suppressed. Furthermore, since a completely filled reference image is used, the problems linked to partial source patches are omitted (*cf.* Section 3.3).

However, as inaccuracies of disparity maps already play a critical role when the patch correspondence search is restricted to the frame itself, depth constraints are only considered at key frames in this thesis. This procedure is based on the assumption that visual disturbances caused by foreground blur are prevented at the very beginning and sound image completion results are then propagated to subsequent frames. In this context, a key frame denotes the first image of a

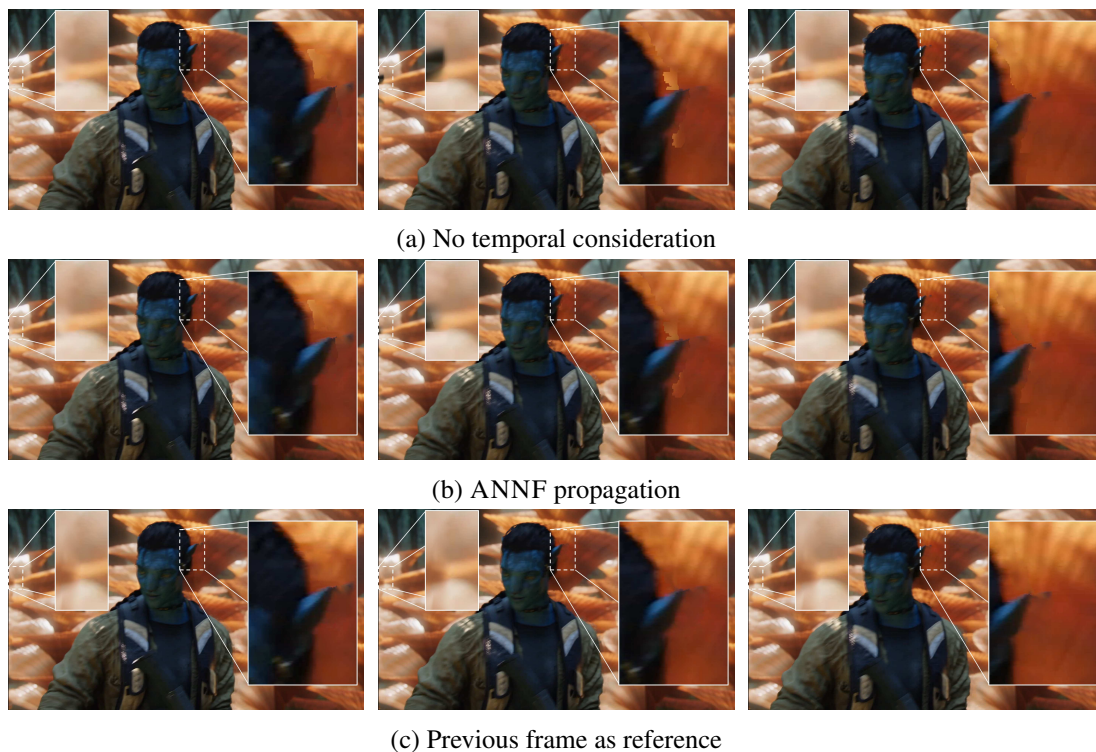


Figure 3.20: Reduction of flickering artifacts. Each row shows the same three consecutive frames, where in (a) no temporal consideration has been given, (b) covers the ANNF propagation approach and in (c) the previous frame is used as the reference image. Note the amount of flickering artifacts, especially in the magnified areas, are significantly reduced in (c) compared to (a).

shot or after a scenecut.

In Figure 3.20, the influence of the different temporal approaches is shown on three consecutive example frames. Note the flickering artifacts caused by the dark spot at the left margin and the bright areas near the head of the character in the middle frame of Figure 3.20(a), which are absent in the first and third frame. Whereas the results of ANNF propagation presented in Figure 3.20(b) show little improvement in the reduction of artifacts, these inconsistencies have almost vanished using the method shown in Figure 3.20(c). Additionally, note that a combination of the two temporal methods can be used to benefit from both computational speedup and quality improvement.

3.9 Additional algorithmic adaptations

In this section, different algorithmic aspects are outlined concerning the employed patch distance metric, the chosen color space and the synthesis of the inpainted pixel color.

3.9.1 Patch distance metric

At the core of every exemplar-based image completion approach lies the search for correspondences between patches of the target region and their best fitting NNs. Formally, for a target patch Ψ_p centered at a pixel p , its corresponding NN Ψ_q is defined as:

$$\Psi_q = \arg \min_{\Psi \in \Phi} D(\Psi_p, \Psi) \quad , \quad (3.7)$$

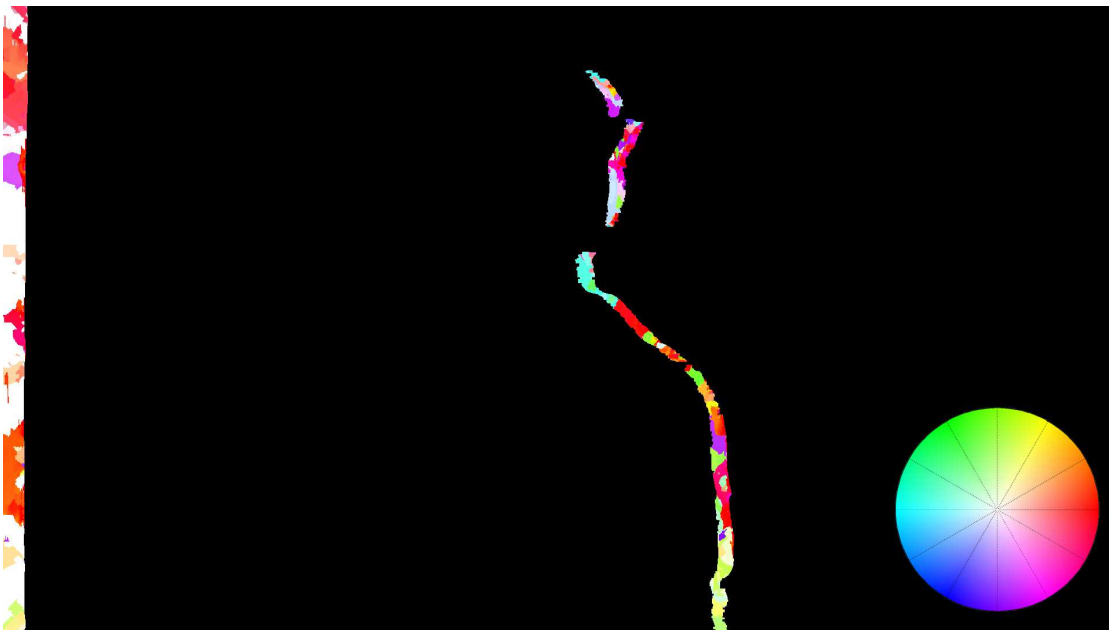
where Φ denotes the source region covering all potential candidate exemplars and $D(\cdot, \cdot)$ refers to a specific patch distance metric. In this thesis, according to its prevalent use in the field of patch matching algorithms [RB12; SN12; HS12], the SSD (as defined in Equation 3.2) is used as the cost function to measure patch similarities. In contrast to the *Sum of Absolute Differences* (SAD), the SSD strongly discriminates large patch differences which results in an enhanced sensitivity to outliers.

However, contrary to existing depth-aided image completion approaches [DP10; MHC+12; AK12], no additional depth-related data term is introduced in the formalism of the cost function. As elucidated in Section 3.7, the depth constraint in the proposed inpainting approach is applied as a strict binary decision boundary, which determines whether a pixel is considered in the matching and inpainting stage or not. Since this constraint already has to be attenuated due to inaccuracies in the underlying disparity maps, an additional consideration in the cost function would entail a preference for certain depth levels. However, this would again lead to a more stringent restriction of valid candidate exemplars, resulting in insufficient inpainting outcomes. Additionally, note that the prevention of foreground blur is not feasible when depth information is only used in the cost function, compared to a strict exclusion of foreground regions as proposed in this thesis.

Similar to a supplementary depth term, also the usage of a spatial extension is omitted. In several works [HB10; HBG11; AK12] this spatial term is applied to encourage candidate exemplars being selected from nearby regions. However, the example shown in Figure 3.21



(a) Inpainting result



(b) Corresponding ANMF

Figure 3.21: Inessentiality of spatial constraints: (a) image completion result including one sample NN pair, where inpainted regions are outlined in red; (b) corresponding ANMF encoded in HSV color space (color wheel included for guidance), where hue refers to the angle and saturation to the inverse distance. White pixels denote patch distances that exceed more than half of the image width (see also text for detailed description).

demonstrates the negligibility of such a spatial constraint. In particular, the frame presented in Figure 3.21(a) exhibits a roughly recurrent background of similar flowers of different sizes. When we take a closer look at the corresponding ANNF after the matching process has finished (see Figure 3.21(b)), one can notice the predominant white areas at the left image border. For the purpose of visualization, the ANNF is encoded in HSV color space where the angle between corresponding patches is illustrated as the hue and the inverse distance represents the saturation. Now, these white pixels indicate NN distances that exceed more than half of the image width. Hence, the large amount of such greater distances refutes the assumption that for disocclusions especially at image borders, which denote newly added areas in the altered field of view of the (virtual) camera, nearby patches should be used to pursue the adjacent boundary regions into the holes. In Figure 3.21(a), one example NN pair is illustrated comprising the target patch (colored in cyan) and the respective source patch (colored in orange).

Consequently, the patch similarity is solely based on color difference in this thesis.

3.9.2 Color space

In the proposed image completion approach, the CIELAB, also referred to as CIE $L^*a^*b^*$, is used as the underlying color space for matching patches. Derived from the CIE XYZ color space, CIELAB is constructed as a perceptual uniform color space defined for the purpose of measuring color differences [KP92]. In contrast to the RGB color space, this perceptual uniformity ensures that the Euclidian distance separating two similar colors is proportional to their visual difference. CIELAB consists of one lightness dimension (L^*) and two chrominance dimensions (a^* , b^*) based on the color opponent process between red - green and yellow - blue. Moreover, this color space is device independent covering all perceivable colors. A lossless transformation to the CIELAB color space via the CIE XYZ tristimulus values is given as follows:

$$\begin{aligned}
 L^* &= 116 \cdot f\left(\frac{Y}{Y_n}\right) - 16 \\
 a^* &= 500 \cdot \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] \\
 b^* &= 200 \cdot \left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right]
 \end{aligned} \tag{3.8}$$

where

$$f(t) = \begin{cases} \sqrt[3]{t} & \text{if } t > \left(\frac{6}{29}\right)^3 \\ \frac{1}{3} \cdot \left(\frac{29}{6}\right)^2 \cdot t + \frac{4}{29} & \text{otherwise} \end{cases} .$$

Here, X_n , Y_n and Z_n denote the CIE XYZ tristimulus values of the reference white point.

3.9.3 Pixel color synthesis

The final step of the proposed image completion pipeline comprises the inpainting of vacant target regions. In this thesis, various approaches have been assessed concerning the actual synthesis of pixel colors. In Figure 3.22, the inpainting results of different synthesis methods are presented in terms of image snippets including the left image border of the sample frame shown in Figure 3.21(a).

A computational fast approach is to simply transfer the color value from the source patch to the correspondent target patch. In particular, for every NN pair solely the central pixel of the source patch is copied to the location of the associated target pixel. However, this pixel-wise synthesis approach disregards the coherence characteristic of images, since every pixel is considered individually. Additionally, recall that the central pixel of a target patch is never taken into account in the matching step as it always constitutes an invalid (i.e. hole) pixel. As can be seen in Figure 3.22(a), these reasons lead to a tremendous number of inconsistencies in the inpainted regions.

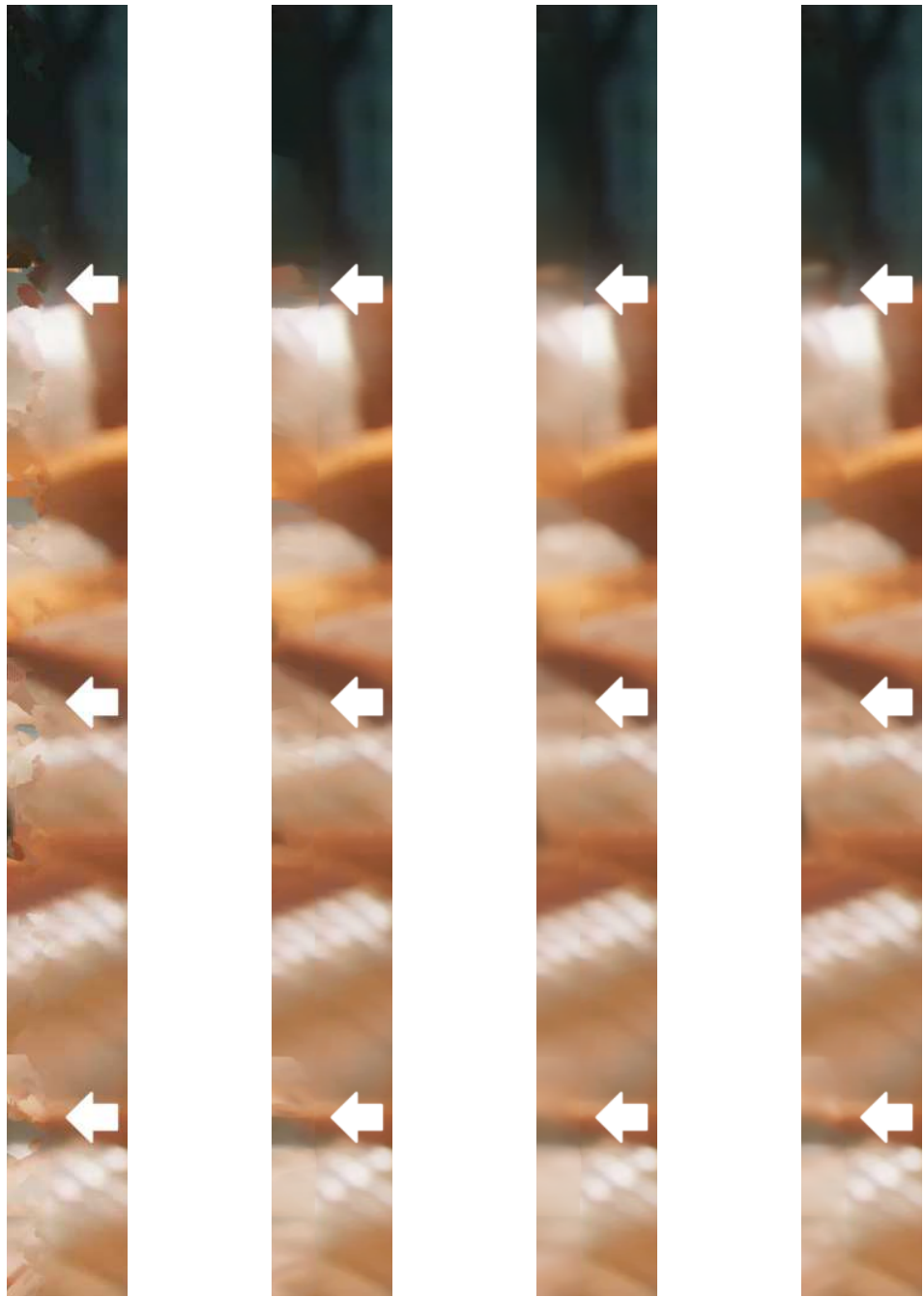
Hence, to account for the natural structure of images, the information of adjacent patches is incorporated in the calculation of the final pixel color, since the output of patch-sampling synthesis algorithms typically contains large chunks of data from the input [A01]. Particularly, a target pixel in a hole region is not only covered by its respective target patch, but also by the matching windows in its vicinity. He et al. [HBG11] proposed to take the median of these overlapping color values at each target location in order to avoid the presence of noise and to prevent high frequencies in the inpainted region. Evidently, the result shown in Figure 3.22(b) exhibits less visual artifacts compared to the pixelwise synthesis method as outliers are filtered out. However, there is a distinct seam discernible at the transition between the inpainted target region and remaining valid parts of the image.

Since the median calculation additionally depicts a computational expensive operation that strongly influences the overall runtime of the image completion system, an alternative approach is to substitute the median by the mean value. However, this leads to a blurrier inpainting result as can be seen in Figure 3.22(c). The blurriness is a consequence that not all overlapping patches agree on the color value of a pixel at one specific location, which causes the final color value to be a mixture of different color shades. Consequently, instead of giving the same influence to pixels that are located, e.g., at the center or at the periphery of a patch, pixel weights are introduced to react to this imbalance.

Hence, the assigned color C of a pixel p at the end of the image completion process is defined as:

$$C(p) = \frac{\sum_{i=1}^n w_i \cdot C(i_p)}{\sum_{i=1}^n w_i} \quad , \quad (3.9)$$

where n is the number of all source patches Ψ^i containing p , $C(i_p)$ is the color of the corresponding pixel in each of the n associated patches Ψ^i and w_i are the respective weights.



(a) Pixelwise

(b) Median overlap

(c) Average overlap

(d) Gaussian overlap

Figure 3.22: Image completion results using different approaches for pixel color synthesis. Note the seam (indicated by the arrows) at the transition between the inpainted target region and the remaining part of the image, which is significantly reduced using (d) Gaussian patch weighting.

In this thesis, to account for adaptive patch sizes and to mitigate the influence of large matching windows on distant target regions, w_i corresponds to Gaussian weights G_i , which are calculated for an $m \times m$ patch by:

$$G_i = \alpha \cdot e^{-\frac{\left(\frac{i-(m-1)}{2}\right)^2}{(2 \cdot \sigma)^2}}, \quad (3.10)$$

where $i = 0 \dots m - 1$, α is the scale factor chosen so that $\sum_i G_i = 1$ and the Gaussian standard deviation is given as $\sigma = \frac{3}{10} \cdot \left[(m - 1) \cdot \frac{1}{2} - 1 \right] + \frac{4}{5}$. The inpainting result of this Gaussian-weighted pixel synthesis method is presented in Figure 3.22(d). As can be seen, the blurriness as well as the seam at the hole boundary are significantly reduced.

3.10 Compared methods

In the final section of this chapter, two image completion approaches are introduced – horizontal background replication and Adobe[®]'s content-aware fill – which are used for comparison in the subsequent evaluation.

3.10.1 Horizontal background replication

Horizontal background replication – also referred to as *copy background* [ESG12] – denotes a simple inpainting method, where visual information in the vicinity of the target region is used to fill in the holes. In particular, the holes are completed by replicating the adjacent image regions into the vacant areas. However, to avoid portions of foreground objects being copied to image gaps caused by disocclusions, depth information is incorporated in the selection of potential source regions.

For an image I the depth constraint of a hole $\mathcal{H} \in \Omega$ is formulated as:

$$I(x, y) = \begin{cases} I(x - \Delta_x, y) & \text{if } d_l > d_r \\ I(x + \Delta_x, y) & \text{if } d_l \leq d_r \end{cases} \quad \forall (x, y) \in \mathcal{H} \quad , \quad (3.11)$$

where Δ_x denotes the size (i.e. the number of pixels) of the respective hole in scanline x and d_l and d_r represent the disparity of the first pixel located at the left and right side of the hole boundary $\delta\mathcal{H}$ according to x .

However, for large target regions, especially at the margin of an image, there may arise the situation where the remaining part of the image will not suffice to fill in the hole completely. Another problem that emerges when considering the infilling of wide gaps is that visual information of additional foreground objects, which are partially or entirely located in the area to be copied, might be taken into account in the completion process.

Thus, not only the depth information at the hole boundary has to be considered, but also the disparity values of the pixels being copied. Eisenbarth et al. [ESG12] proposed to abort the copy procedure if the difference of the disparities between the boundary pixel (i.e. d_l or d_r) and the currently copied pixel exceeds a predefined threshold. The remaining part of the hole is then completed by repetitive infilling of already used valid image portions. The image completion results of three sample frames are shown in Figure 3.25(a).

3.10.2 Adobe Photoshop: content-aware fill

Compared to horizontal background replication, Adobe[®]'s content-aware fill depicts a more sophisticated inpainting technique. As stated in [AR10], this image completion approach, which is available since Photoshop[®] CS5, is based on the PatchMatch algorithm of Barnes et al. [BSF+09] and the hole filling method proposed by Wexler et al. [WSI07]. Since the PatchMatch algorithm has been discussed in detail in Section 3.1, now the deviations of Wexler et al.'s method to the proposed inpainting approach are addressed subsequently.

Since Photoshop[®] is a commercial graphics editing program, there exists no literature that describes the exact functional principles of their hole filling method. Consequently, as the work in [WSI07] was originally proposed for the completion of missing information in video sequences using 3-dimensional space-time patches, but on the other hand, Photoshop[®] is mainly designed for processing 2-dimensional content, the temporal aspects of Wexler et al.'s approach are omitted in the following discussion.

In [WSI07], the image completion process is formulated as a global optimization problem to ensure visual coherence between the synthesized data Ω^* in the target region Ω and the pre-existing image portions in the source region Φ . Hence, the aim is to determine visual data Ω^* which maximizes the following objective function:

$$Coherence(\Omega^*|\Phi) = \prod_{p \in \Omega^*} \max_{q \in \Phi} sim(\Psi_p, \Psi_q) \quad , \quad (3.12)$$

where p and q run over all pixel positions in their respective (target or source) region. The local similarity measure $sim(\cdot, \cdot)$ between two patches Ψ_p and Ψ_q is defined as:

$$sim(\Psi_p, \Psi_q) = e^{-\frac{D(\Psi_p, \Psi_q)}{2\sigma^2}} \quad , \quad (3.13)$$

where the distance metric $D(\cdot, \cdot)$ is specified as the SSD of RGB values and σ , which controls the smoothness of the induced error surface, is set to be the 75-percentile of all distances in the current search in all locations.

Wexler et al. stated that Equation 3.12 will be satisfied if the following two conditions are met at every target pixel p :

1. All target patches $\Psi_{p_1} \dots \Psi_{p_k}$ containing p appear in the source region Φ :

$$\exists \Psi^i \in \Phi, \Psi_{p_i} = \Psi^i$$

2. All these corresponding source patches $\Psi^1 \dots \Psi^k$ agree on the color value C at location p :

$$C(p) = \Psi^i(p) = \Psi^j(p)$$

Now, the main difference between the work in [WSI07] to the proposed inpainting framework is that Wexler et al. gradually put confidence into newly synthesized image data. Compared to the proposed onetime infilling approach illustrated in Figure 3.4, this leads to a repetitive inpainting step where color values in the target regions are updated at every iteration. Figure 3.23

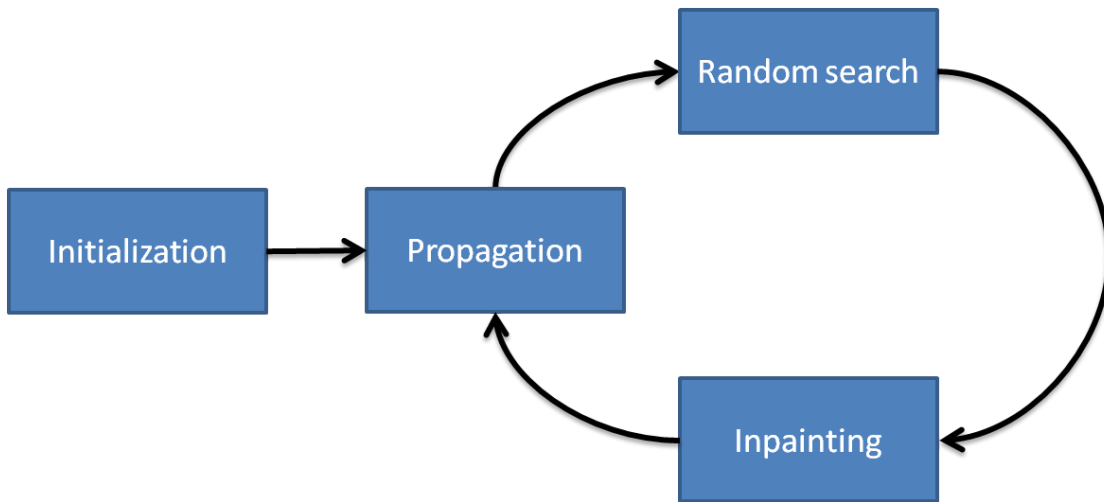


Figure 3.23: Workflow of PatchMatch-based image completion using iterative inpainting.

visualizes the overall workflow. Here, the color values are likewise calculated as defined in Equation 3.9. However, instead of using Gaussian weights, Wexler et al. choose the similarity measure specified in Equation 3.13 as their weights, which measures the degree of reliability of target patches according to Condition 1. Thus, the most likely color C at location p minimizes the variance of the colors $C^1 \dots C^k$ proposed by $\Psi^1 \dots \Psi^k$ to satisfy Condition 2. In particular, the weights are defined as $w_{i_p} = \alpha_p \cdot sim(\Psi_{p_i}, \Psi^i)$, where α_p denotes an additional confidence term at each pixel, where known points in the source region have a fixed high confidence and

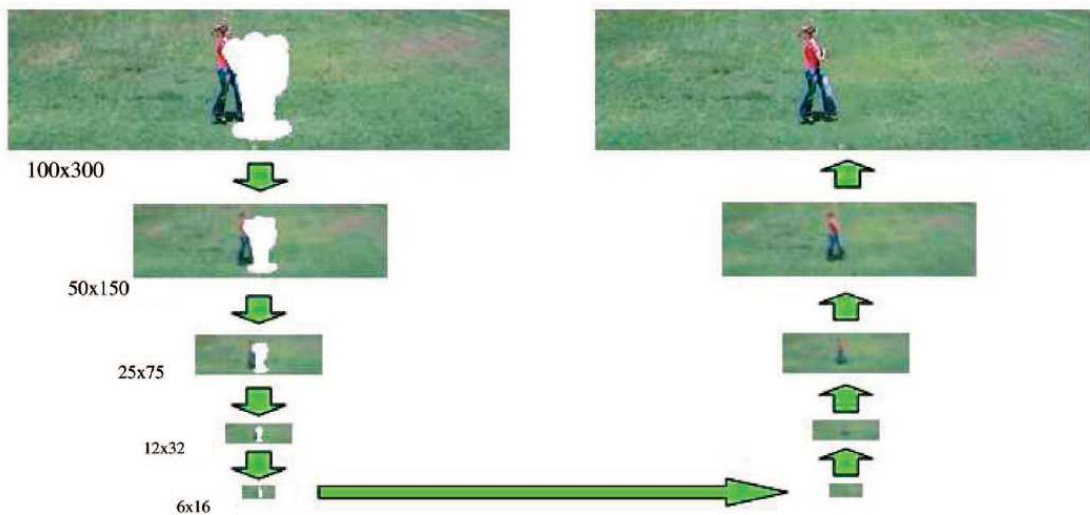


Figure 3.24: Multiscale approach. The algorithm starts by shrinking the input image. The completions starts at the coarsest level, iterating each level several times and propagating the result upward [WSI07].

the confidence of target pixels is attenuated according to their distance to the hole boundary. Additionally, the Mean-Shift algorithm [CM02] is used to reduce the sensitivity to outliers.

Since in this repetitive inpainting approach initialization depicts a crucial step as the newly synthesized data impact the subsequent NNS, Wexler et al. perform the iterative process in multiple scales using image pyramids as can be seen in Figure 3.24, where each pyramid level contains half the resolution. The optimization starts at the coarsest octave and the solution is propagated to finer levels for further refinement. After propagation of source patch locations onto the finer level, instead of plain interpolation, only those patches that still overlap the target pixel are used for the pixel color synthesis to avoid unnecessary blur. Additionally, this pyramid approach speeds up convergence and accounts for blanks (see Section 3.3) caused by fixed patch sizes as the number of hole pixels (but also that of source pixels) is reduced at every octave. The image completion results of three sample frames using content-aware fill are shown in Figure 3.25(b).

Note that in the proposed image completion framework global coherence is encouraged by the propagation step of the PatchMatch algorithm and the influence of outliers on the inpainted color values is attenuated by using Gaussian patch weights.

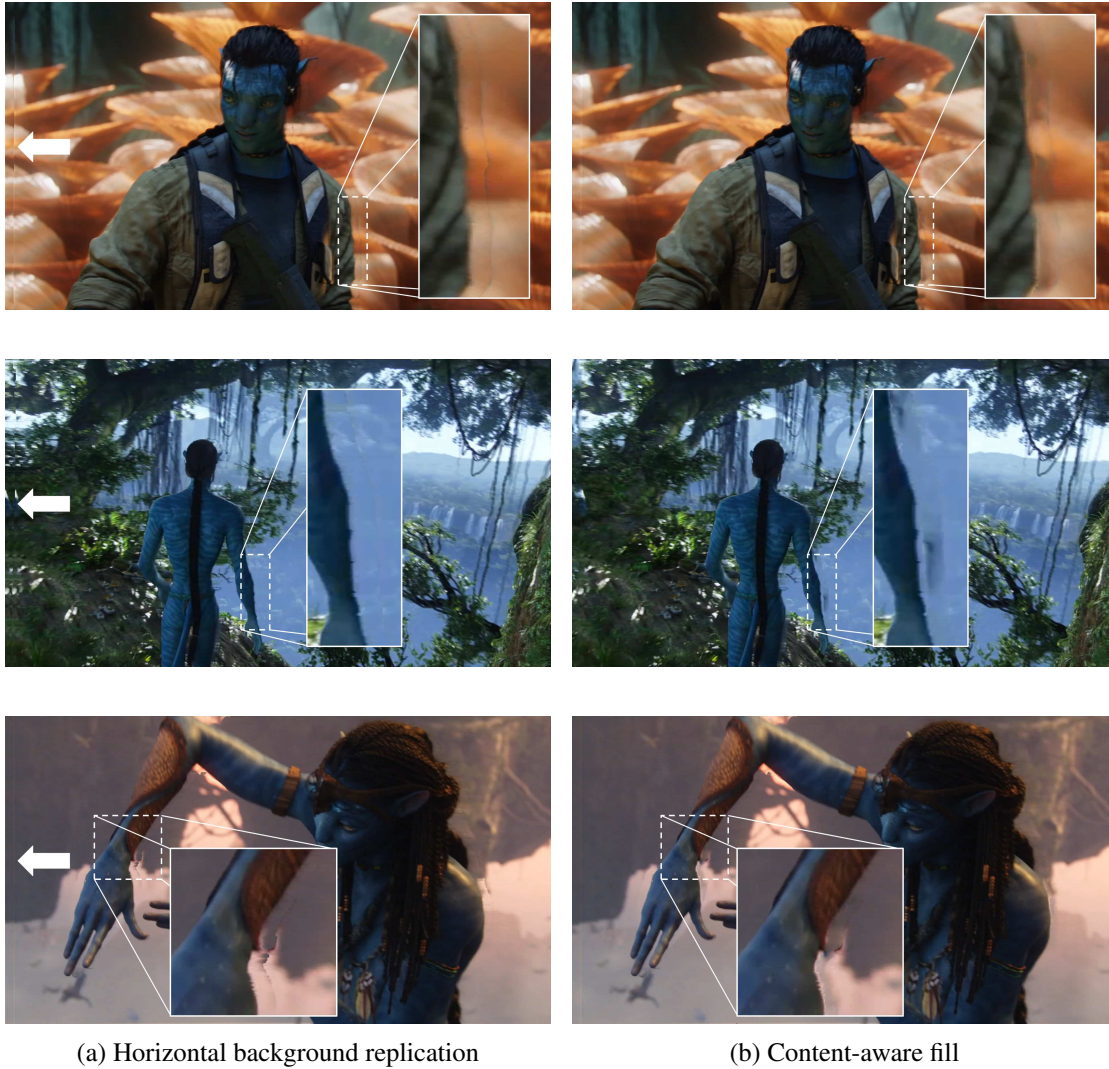


Figure 3.25: Image completion examples using compared inpainting approaches. Note the impact of warping artifacts and foreground blur on the visual impression of the infilling results. A distinct seam (indicated by the arrows) can be seen at the boundary of the inpainted region on the left margin of the image when horizontal background replication is applied. In the proposed inpainting approach these artifacts are eliminated using single-sided mask dilation and depth constraints (see Section 3.4.1 and 3.7).

Evaluation and Results

In the following chapter, the results of the evaluation are presented and discussed in detail. Regarding the quality of the completed images, the different enhancement stages of the proposed inpainting approach are compared to “*horizontal background replication*” and the image completion function *content-aware fill* in Adobe®’s Photoshop® CS5. To evaluate both the depth-based and the temporal extensions of the presented framework, the inpainting methods are applied to still images as well as video sequences. Moreover, the evaluation of the inpainting results is conducted with objective quality metrics as well as a subjective user study (*cf.* Appendix A). The used dataset and the implementation details are presented in addition.

4.1 Database

All inpainting methods are evaluated on footage selected from the 2009 movie *Avatar*, which is the most popular and highest-grossing 3D motion pictures in the world¹. The test frames are obtained by warping the left camera image onto the position of the right one. Consequently, this experimental setup allows to use the original right camera frame as the underlying *Ground Truth* (GT) data, whose existence is a necessary precondition for the calculation of objective quality metrics.

In particular, five still images as well as five video sequences – termed as *Bird*, *Flower*, *Edge*, *Arm* (image) / *Couple* (video) and *Crowd* – have been chosen as illustrated in Figure 4.1 and Figure 4.2, respectively. Moreover, both the selected still images and the chosen video sequences cover different image characteristics including varying densities of background texture and diverse amounts of target pixels, as summarized in Figure 4.1(f) and Table 4.1. Each frame features a *Full High Definition* (Full HD) resolution of 1920×1080 pixels. Hence, the span of the number of target pixels to be filled in, which ranges from 29790 to 57711, corresponds to a percentage of 1.4 and 2.7 of the total amount of image pixels. Note that this number increases

¹<http://www.boxofficemojo.com/alltime/world/>, accessed: 2013-11-15



Figure 4.1: Input frames for evaluation on still images. Black image borders are added to highlight the delimitation of the frames (note the gaps at the left side).

by about 0.5% (10368 pixels) for the proposed inpainting approaches that employ the additional single-sided dilation preprocessing step.

The evaluated methods in conjunction with their used abbreviations are summarized in Table 4.2. Besides the initial PatchMatch algorithm (PM), its presented adaptations (APM) and its proposed depth (DPM) and temporal (AP, PF) extensions, also the computationally efficient horizontal background replication (HBR) and the image completion function content-aware fill (CAF) of Adobe[®]'s Photoshop[®] CS5, which is the prevalent image editing application used by over 90% of graphics professionals², are compared.

²<http://www.adobe.com/se/company/fast-facts.html>, accessed: 2013-11-6.

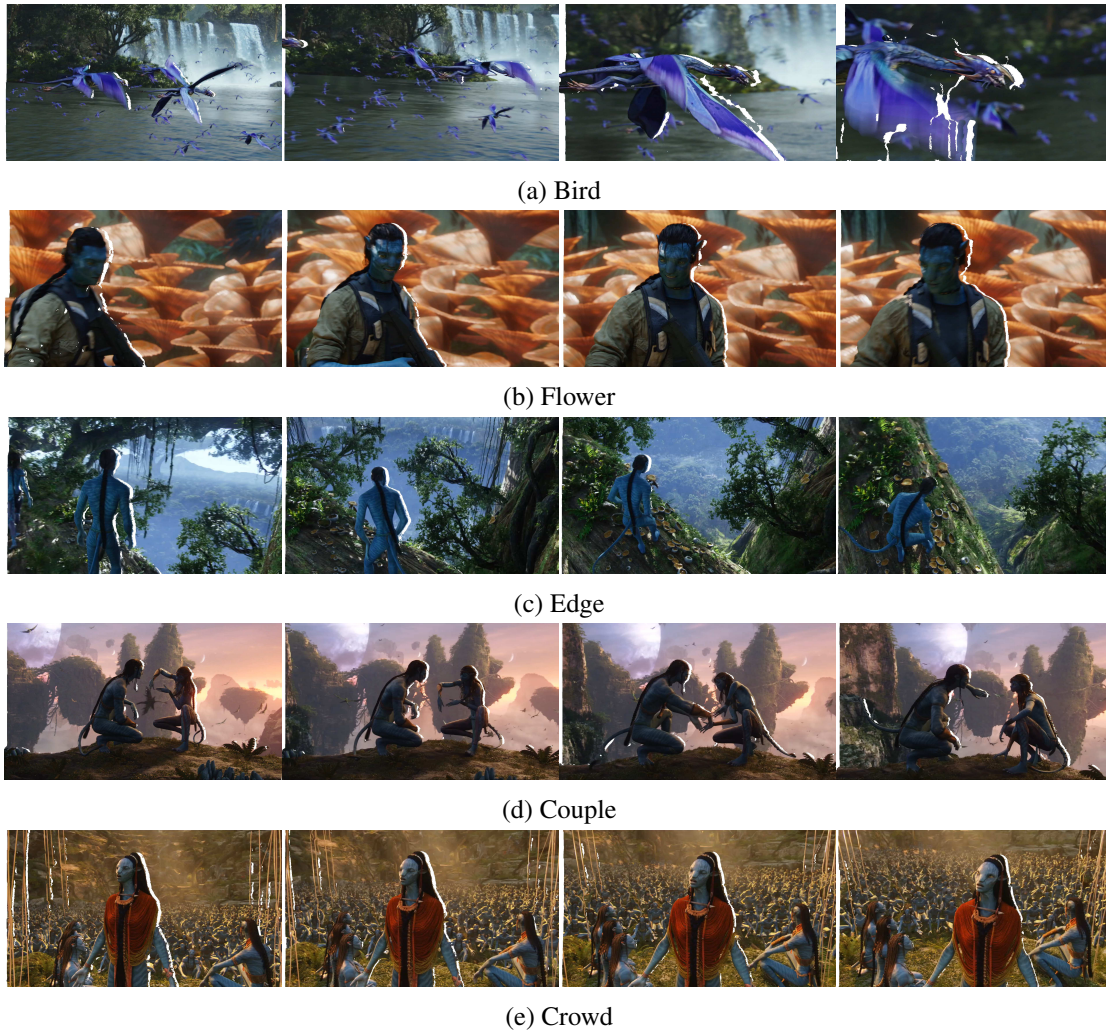


Figure 4.2: Input sample frames of the video sequences used for evaluation.

NAME	# FRAMES	# TARGET PIXEL	CHARACTERISTIC
Bird	112 (4'')	51728 (2.5%)	increased camera motion
Flower	60 (2'')	46829 (2.2%)	clear repetitive background
Edge	159 (6'')	34004 (1.6%)	highly textured background
Couple	127 (5'')	33515 (1.6%)	moderate textured background
Crowd	136 (5'')	51841 (2.5%)	cluttered repetitive background

Table 4.1: Specification of video sequences including the number of frames, the runtime in seconds ("), a brief description of the footage's characteristic and the average number of target pixels per video.

NAME	DESCRIPTION	SECTION
PM	Initial PatchMatch using fixed patch sizes (51 pixels), uniform-weighted pixel synthesis and RGB color space	3.1-3.2
APM	Adapted PM including additional preprocessing steps, adaptive patch sizes, matching pixel threshold, Gaussian-weighted pixel synthesis and CIELAB	3.4-3.6,3.9
DPM	APM plus depth extension	3.7
CAF	Content-aware fill	3.10.2
HBR	Horizontal background replication	3.10.1
NT	No temporal consideration (plain DPM)	—
AP	ANNF propagation	3.8
PF	Previous frame as reference	
GT	Ground truth (original right view)	—

Table 4.2: Description of the used abbreviations for the evaluated inpainting methods.

All experiments are executed on an Intel®Xeon® W3520 2.66 GHz processor including 12 GB RAM. The proposed algorithms are implemented in C++ using OpenCV 2.4 library functions [B00].

4.2 Objective metrics

To evaluate the performance of the presented image completion approaches in the inpainted regions, three error measures are selected that are widely used in the field of computer vision. The *Root Mean Square Error* (RMSE) represents the radical of the cumulative squared error between the inpainted image I_{ip} and the original image I_{gt} of the same size:

$$RMSE = \sqrt{\frac{\sum (I_{gt} - I_{ip})^2}{w \cdot h \cdot 3}}, \quad (4.1)$$

where w and h are the width and height of the images, respectively.

The term *Peak Signal-to-Noise Ratio* (PSNR) is an expression for the ratio between the maximum possible power (value) of a signal and the power of corrupting noise that affects the fidelity of its representation. In this case, the signal represents the original image and the noise is the error introduced by inpainting. Since many signals have a wide dynamic range, PSNR is typically defined in terms of the logarithmic decibel (dB) scale [AN12] as:

$$PSNR = 20 \cdot \log_{10} \left(\frac{MAX}{RMSE} \right), \quad (4.2)$$

where MAX corresponds to the maximum fluctuation according to the image data type (e.g. $MAX = 255$ in case of an 8-bit representation per color channel). PSNR takes values from 0

to infinity, where a higher score typically indicates a better inpainting result. Average values in the field of image completion are between 20 dB and 30 dB, but highly depend on the utilized content [SLW+03; SC05; XS10].

Since RMSE and PSNR do not reflect the behavior of the human visual system, Wang et al. [WBS+04] attempt to estimate the perceptual quality of an image as the perceived change in structural information. The so-called *Structural SIMilarity* (SSIM) index is formulated as a full reference image quality metric that measures the quality of a candidate patch Ψ_{ip} with respect to the original image patch Ψ_{gt} of the same size:

$$SSIM(\Psi_{gt}, \Psi_{ip}) = \frac{(2\mu_{\Psi_{gt}}\mu_{\Psi_{ip}} + C_1) \cdot (2\sigma_{\Psi_{gt}\Psi_{ip}} + C_2)}{(\mu_{\Psi_{gt}}^2 + \mu_{\Psi_{ip}}^2 + C_1) \cdot (\sigma_{\Psi_{gt}}^2 + \sigma_{\Psi_{ip}}^2 + C_2)}, \quad (4.3)$$

where $\mu_{\Psi_{gt}}, \mu_{\Psi_{ip}}$ and $\sigma_{\Psi_{gt}}, \sigma_{\Psi_{ip}}$ are the mean and the variance of the patches Ψ_{gt} and Ψ_{ip} , respectively. $\sigma_{\Psi_{gt}\Psi_{ip}}$ denotes the covariance between Ψ_{gt} and Ψ_{ip} . $C_1 = (k_1L)$ and $C_2 = (k_2L)$ are used to stabilize the division by weak denominators, where L indicates the range of pixel values (usually $2^{\#bitsperpixel} - 1$) and k_1, k_2 are set to 0.01, 0.03 by default.

To compute the overall image quality, the *mean* of the SSIM (MSSIM) index is used:

$$MSSIM(I_{gt}, I_{ip}) = \frac{1}{N} \sum_{i=1}^N SSIM(\Psi_{gt_i}, \Psi_{ip_i}), \quad (4.4)$$

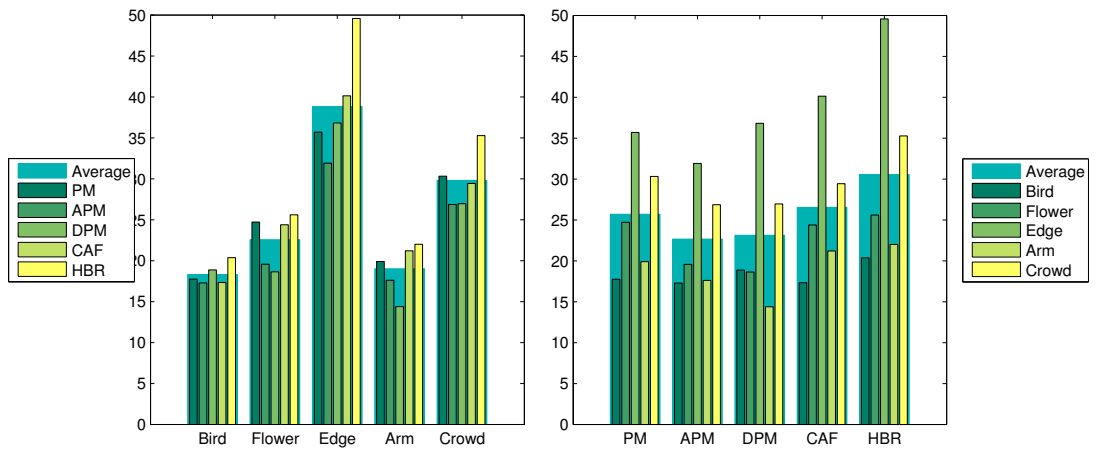
where N is the total number of patches and Ψ_{gt_i} and Ψ_{ip_i} are the i -th patches of images I_{gt} and I_{ip} , respectively. The resultant MSSIM value³ takes values between -1 and 1 , where 1 indicates the case of two identical images.

Note that the final scores for all error measures previously mentioned are averaged over all RGB color channels. Alternatively, the quality metrics can be computed only on the luminance channel by converting the color images to a different color space (e.g. YCbCr) to account for the sensitivity of the human eye to intensity information [AN12]. However, experiments have shown that the beneficial effect is negligible.

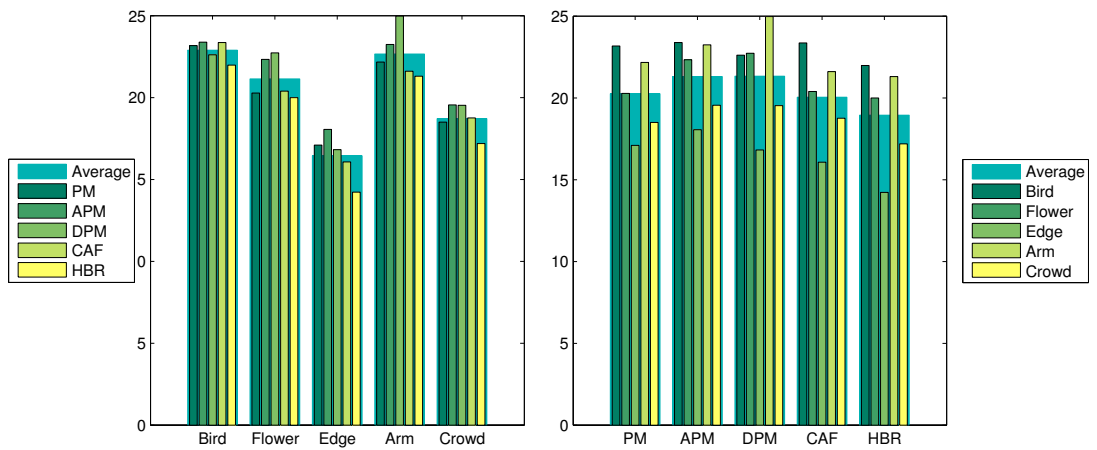
4.2.1 Still images

In Figure 4.3, the evaluation results of the selected images are shown according to presented quality metrics. It can be seen that the average RMSE of image *Edge* exceeds those of *Bird* and *Arm* by more than 19 units. This is directly related to the fact that *Edge* exhibits a considerably higher amount of texture details. Moreover, both PatchMatch-based inpainting approaches PM and CAF attain approximately similar RMSEs of 25.66 and 26.50. The proposed PatchMatch adaptations APM and DPM, nevertheless, eventuate in the smallest errors of 22.64 and 23.13, respectively. Additionally, HBR yields the highest RMSE because it preserves fine structures of the texture as image portions are simply copied, which is contrary to e.g. PM that produces blurrier inpainting results caused by uniform-weighted pixel synthesis (*cf.* Figure 3.25). Consequently, this puts into question the reliability of such objective metrics in the field of image quality assessment.

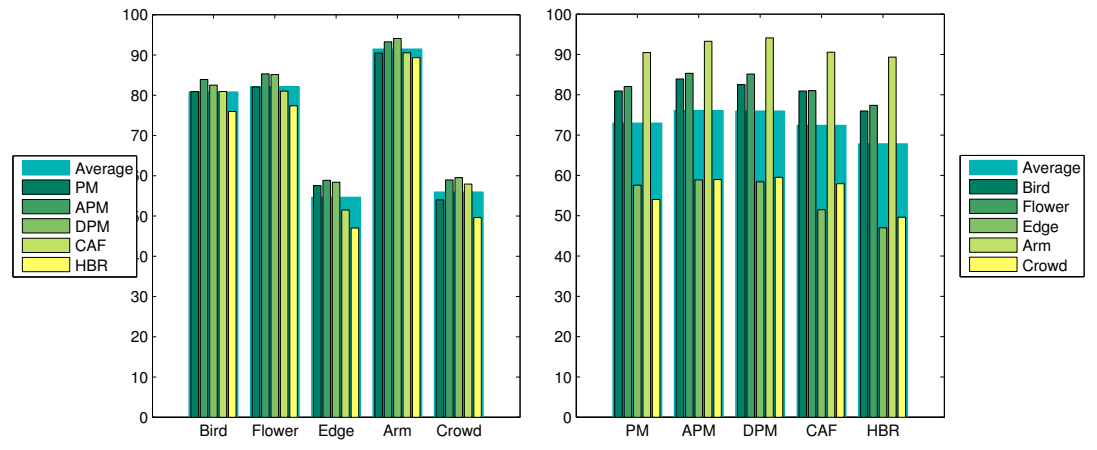
³Note that in the subsequent discussion the MSSIM scores are scaled by the factor 100.



(a) RMSE



(b) PSNR



(c) MSSIM

Figure 4.3: Objective image quality assessment: per image (left) and per method (right).

When the outcomes of Figure 4.3(a) and Figure 4.3(b) are compared, the correlation between RMSE and PSNR becomes apparent. Clearly, a high RMSE results in a low PSNR and vice versa, but the scores are located more closely due to the underlying logarithmic scale. Moreover, note that the inpainting approach with the highest rating varies per image between APM (*Bird, Edge*) and DPM (*Flower, Arm*) where the latter reaches the absolute maximum value at a PSNR of 24.97. This may indicate that the additional use of depth information depends on the particular image content. However, as mentioned before, the validity of these error measurements is limited in this regard.

As can be seen in Figure 4.3(c), the MSSIM index also reflects the leading position of the proposed inpainting approaches DPM (75.90) and APM (76.01), solely their ranking for *Flower* and *Crowd* has swapped. The large variation of the results between the individual images, e.g. 58.39 (*Edge*) and 94.04 (*Arm*) for DPM, shows the dependence to the underlying image content. Moreover, the stronger weighting of structure in the MSSIM index compared to PSNR is discernible by the lower score for *Crowd* and the higher value for *Flower*. However, the difference between the scores of the compared methods using objective quality metrics are small and erroneously suggest minor differences in the quality of the inpainting results.

4.2.2 Video sequences

Since temporal inconsistencies in the inpainted regions tend to be more disturbing to an observer than varying filling accuracy regarding spatial information, another quality measure – besides the aforementioned objective image metrics – is introduced to quantify the amount of flicker in a video sequence. For that purpose, the metric of Schmeing and Jiang [SJ11] has been adapted which determines the quantity of flicker at a fixed pixel position p by counting the cracks in its sequence of color values. However, Schmeing and Jiang performed their assessment only on synthetic videos, where both appearance and location remain constant throughout the sequence, but disocclusions mainly occur near the boundaries of foreground objects that typically change their position and shape over time (e.g. a walking person).

Hence, to account for real world videos in this thesis, the *Relative Frame Differential Flicker* (RFDF) is defined as the average $\kappa(t)$ value over all frames T :

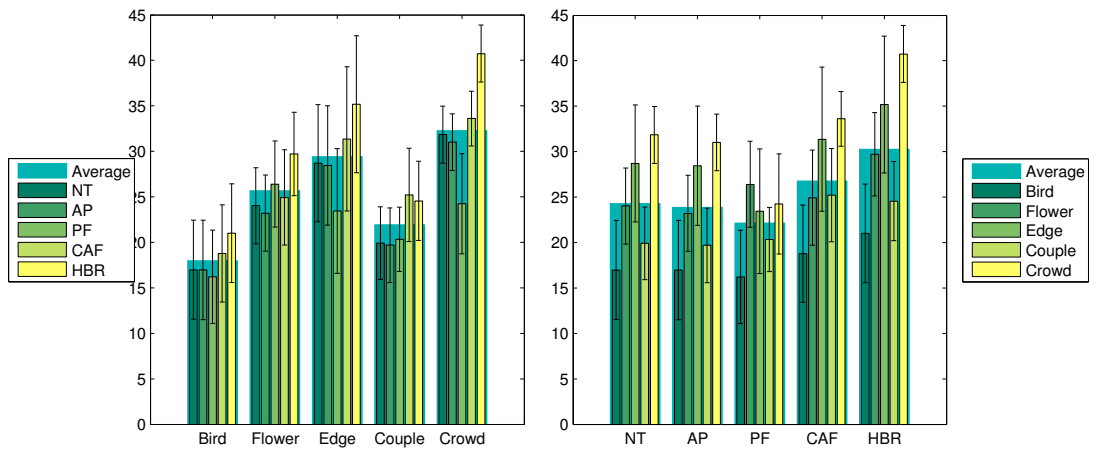
$$RFDF = \frac{1}{T} \sum_{t=1}^T (\kappa(t)) \quad , \quad (4.5)$$

where $\kappa(t)$ is specified as the difference of subsequent reconstructed frames I_{ip} in relation to the difference of their corresponding original frames I_{gt} at time t and $t - 1$:

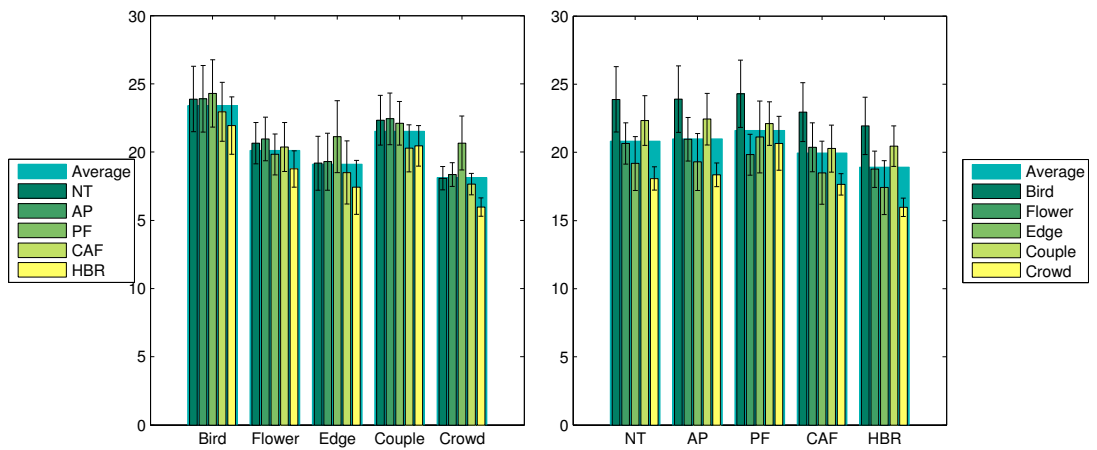
$$\kappa(t) = \frac{1}{N} \sum_{p=1}^N \left(\frac{|I_{ip}^t(p) - I_{ip}^{t-1}(p)|}{|I_{gt}^t(p) - I_{gt}^{t-1}(p)|} \right) \quad , \quad (4.6)$$

where N is the total number of pixels in the current target region.

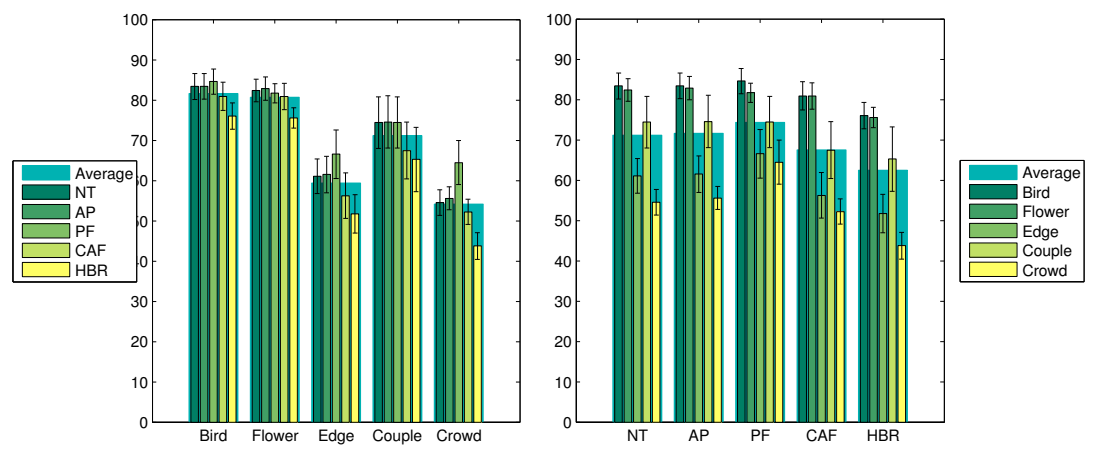
In Figure 4.4, the average RMSE, PSNR and MSSIM values over all frames per video sequence are shown in conjunction with their respective standard deviation. As can be seen from comparison with Figure 4.3, the average RMSE of sequence *Edge* is about 9.40 units smaller



(a) RMSE



(b) PSNR



(c) MSSIM

Figure 4.4: Objective video quality assessment: per sequence (left) and per method (right).

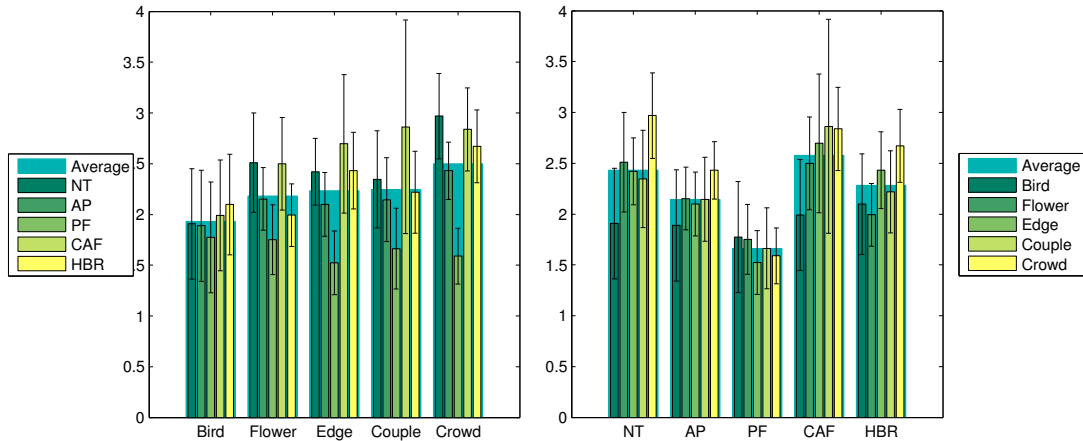


Figure 4.5: RFDF scores: per sequence (left) and per method (right).

compared to its still image counterpart. This is caused by strong camera motion and movement of the foreground object coupled with highly textured background areas, which is additionally indicated by the highest standard deviation of 7.94 among all sequences. Furthermore, AP solely provides a similar score as the non-temporal approach NT. This is a result of the motion occurring in the sequences that cause correspondences of the propagated ANNF to be skipped during the NNS of the subsequent frame due to invalid pixel locations or depth ranges. The proposed temporal inpainting technique PF with an average error of 22.13 outperforms NT by 2.16, showing its merits especially in highly textured sequences (*Edge*, *Crowd*), whereas CAF and HBR deliver similar results of 26.78 and 30.24 related to their still image RMSEs. The findings regarding RMSE applies also to PSNR and MSSIM.

In Figure 4.5, the respective RFDF scores are illustrated. As was to be expected, the methods that do not use any temporal consideration perform poorly. In particular, CAF yields the highest average RFDF value of 2.58. Its large deviation of 0.63 may reflect the random- and approximation-based nature of the PatchMatch algorithm as the consistency of the image completion results can vary significantly from frame to frame. This randomness is reduced in the inpainting results of NT (2.43) which incorporates the depth-related restriction of the NN search space. The lowest score of 1.66 is achieved by PF that – as stated before – demonstrates its strength most notably in the highly textured sequences *Edge* and *Crowd*. However, compared to AP (2.14), HBR has a higher RFDF of 2.28 even though HBR suppresses flickering by copying nearby image parts to fill the holes. This once again reveals the drawback of such pixelwise error metrics in the field of image and video quality assessment.

4.3 Subjective user study

The lack of adequate objective image and video quality assessment metrics – including the field of stereo vision – has been addressed in several scientific papers [HWD+08; WYY+09; WBB+13]. Experimental results have shown that the outputs of these objective measures de-



Figure 4.6: Subjective user study.

pend significantly on the underlying content and correlate poorly with the human perception of quality. Consequently, a supplementary subjective user study has been conducted to verify the previously calculated objective results.

According to the recommendation *BT.2021* of the *International Telecommunication Union* (ITU) proposed in [ITU+12], the *Pair Comparison* (PC) method has been chosen to quantify the subjective ratings. In the PC method, the set of inpainting algorithms is compared in pairs (i.e. two at the time) for each still image and video sequence. The viewers are asked to make a judgment which element in a pair is preferred in the context of the test scenario using a ternary scale (i.e. A is preferred, B is preferred or equally preferred). Particularly, for each precedence of an inpainting method, its respective counter is increased by 1 or 0.5 in case of an equal valuation. The accumulated value is then divided by the number of comparisons per method and by the total number of participants⁴. Hence, the final score shows the percentage of comparisons “won”, e.g. a value of 100 indicates that this method has always been preferred over any other approach.

The inpainting methods under tests are arranged in all possible $\frac{n \cdot (n-1)}{2}$ combinations, where each pair is presented successively on the same display in random order. Since the user can switch interactively between the two elements in a pair, only one of the two possible orders is displayed (i.e. AB or BA), thus the number of required judgments is halved. In this thesis, using 6 inpainting approaches and 5 still images and video sequences, results in a total number of $\frac{6 \cdot (6-1)}{2} \cdot 5 \cdot 2 = 150$ comparisons corresponding to an average test duration of 40 minutes. The experiment starts with a short training stage to familiarize the participants with the upcoming visual disturbances. Additionally, each trail is initiated by the presentation of a mid-gray field

⁴Note that in the subsequent discussion the PC scores are scaled by the factor 100.

containing the trail number at zero disparity.

The test sequences are displayed on a 23.6" Acer GD245HQ monitor featuring a 1920 × 1080 HD pixel grid at a refresh rate of 120 Hertz (Hz), a brightness of 300 candela per square meter (cd/m^2), a contrast ratio of 80000:1 and a response time of 2 milliseconds (ms). The three-dimensional representation is accomplished by the NVIDIA[®] 3D Vision[™] Kit comprising wireless shutter glasses and a USB infrared emitter. To provide an ideal test setup, the room was darkened to avoid external visual disturbances and the viewing distance was set to one and a half times the screen size.

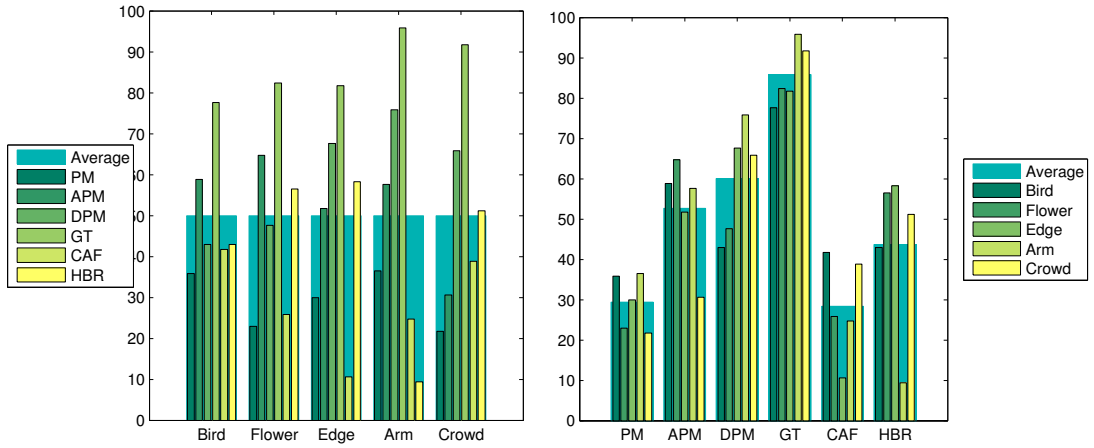
Seventeen non-expert observers (six female and eleven male observers aged between 17 and 49) that had prior experience with three-dimensional content volunteered to participate in the experiment. All participants have been tested on visual acuity (Snellen chart), color perception (Ishihara test) as well as fine and dynamic stereopsis (chart VT-04 and VT-07 in [ITU+12]).

4.3.1 Still images

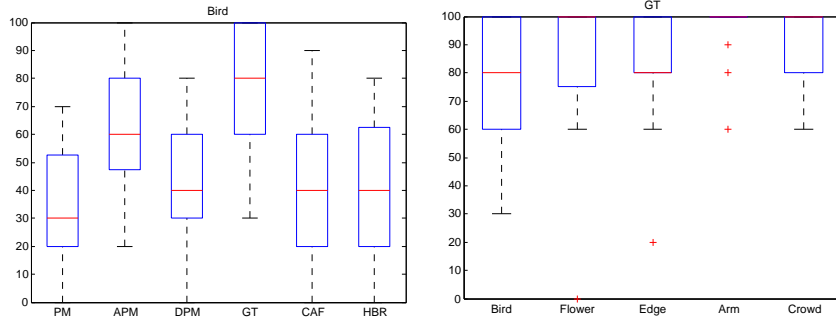
In Figure 4.7, the PC scores of the selected still images are presented. As can be seen, the GT data has clearly been identified by the observers and is preferred in 85.89% of all comparisons. This is mainly caused by the occurrence of warping artifacts that lead to a diminished quality of the overall visual impression. More precisely, these warping artifacts not only degenerate the performance of the image completion algorithms but also provoke visual interferences outside the target regions, e.g. note the slight deformations in the vines at the edges of the scene shown in Figure 4.8. Moreover, GT yields a maximum PC score of 95.89 for the image *Arm* as the disturbance near the thumb of the character shown in Figure 4.9 is located almost at the center of the image and thus in the main focus of the observer's attention.

Similar to the objective quality metrics, both PatchMatch-based inpainting approaches PM and CAF attain approximately equal PC scores of 29.41 and 28.35, respectively. However, contrary to the objective assessment, HBR achieves a significantly better result (43.65) than CAF and even surpasses APM for images *Edge* and *Crowd*. The reason therefore may lie in the fact that in the used test images the inconsistencies caused by HBR inpainting become primarily noticeable at highly textured background regions near the image margin, whereas observers pay more attention to the central area covered by the foreground object. This once more shows that for the assessment of visual quality of images or videos various aspects have to be taken into consideration.

Furthermore, compared to the ranking of e.g. MSSIM index, here the superior performance of the DPM approach towards the results of APM distinctly becomes apparent. In this regard, the study participants remarked a clearer delineation of the foreground objects, which results from the reduction of artifacts caused by foreground blur that are mainly perceived as unnatural shadows of the objects. Striking, however, is the lower score of DPM (47.65) for the image *Flower* compared to APM (64.71) and HBR (56.48). As can be seen in Figure 4.10, a possible explanation for the poor rating may be the bright spot located next to the head of the character as differences in lightness are found particularly disturbing. This visual interference is caused by inaccuracies of the corresponding depth map, since parts of the background have been erroneously considered as foreground and thus are not taken into account in the matching step

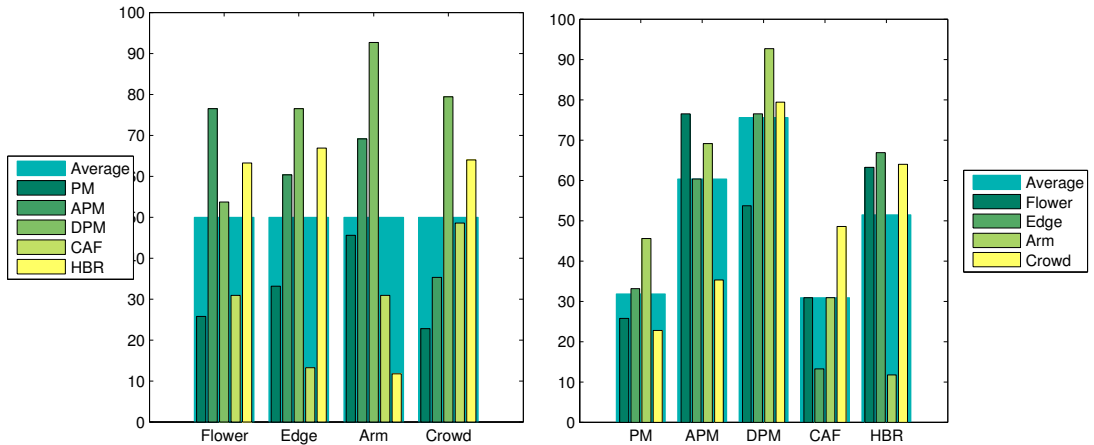


(a) PC scores: per sequence (left) and per method (right).



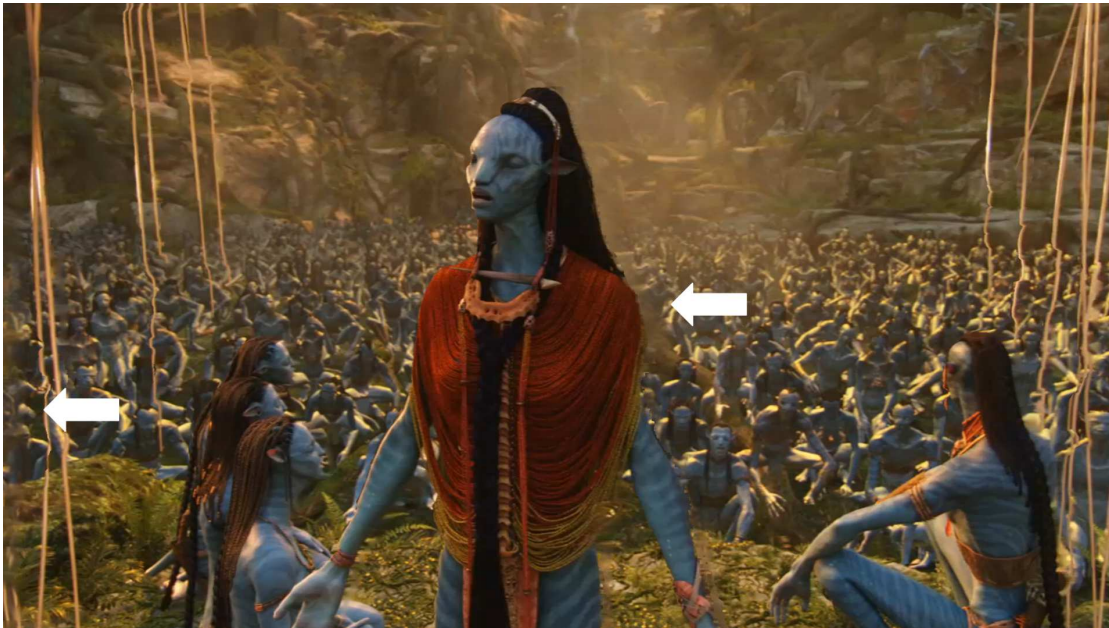
(b) Box plots for *Bird* (left) and *GT* (right).

On each box, the red line indicates the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered as outliers (red crosses). Points are drawn as outliers if they are larger than $q_3 + 1.5 \cdot (q_3 - q_1)$ or smaller than $q_1 - 1.5 \cdot (q_3 - q_1)$, where q_1 and q_3 are the 25th and 75th percentiles, respectively.

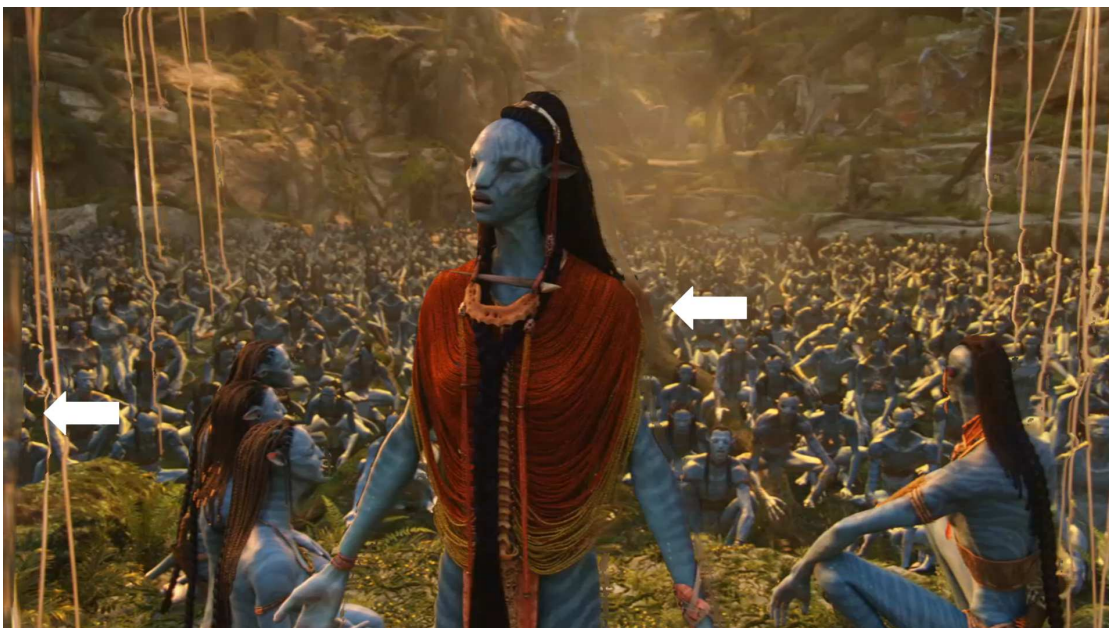


(c) PC scores discarding *Bird* and *GT*.

Figure 4.7: Pair comparison scores for still images.



(a) DPM

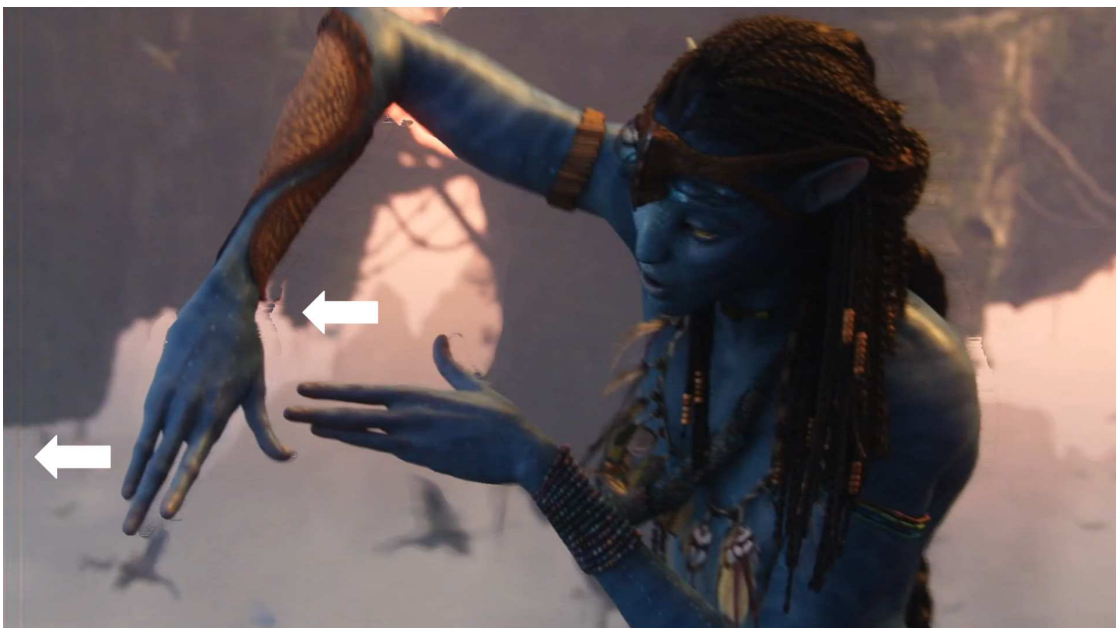


(b) PM

Figure 4.8: Comparison of inpainting results for image *Crowd*: DPM versus PM.



(a) DPM



(b) HBR

Figure 4.9: Comparison of inpainting results for image *Arm*: DPM versus HBR.

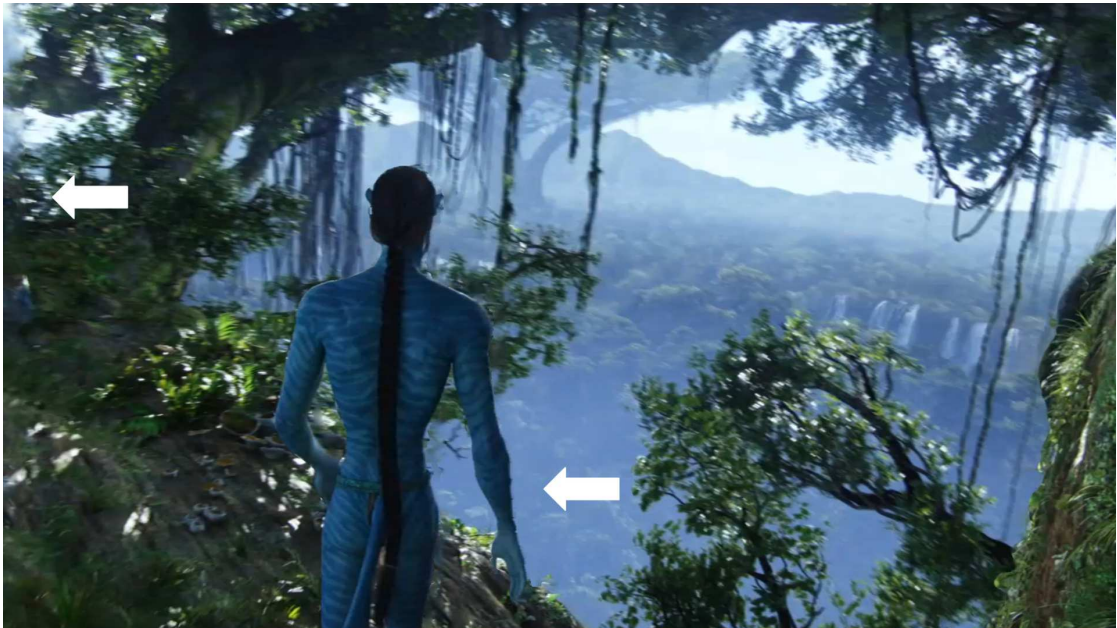


(a) DPM

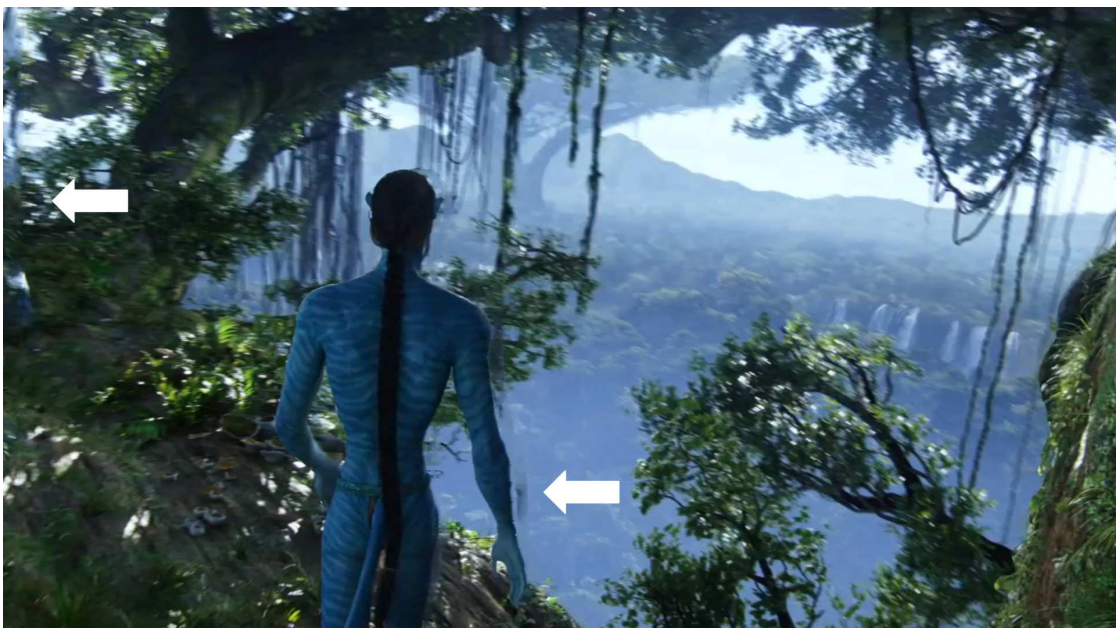


(b) APM

Figure 4.10: Comparison of inpainting results for image *Flower*: DPM versus APM.



(a) DPM



(b) CAF

Figure 4.11: Comparison of inpainting results for image *Edge*: DPM versus CAF.

according to the predefined depth constraints. However, these kinds of inpainting artifacts can be avoided by adjusting the scaling factor α in the depth-based outlier removal (see Section 3.7).

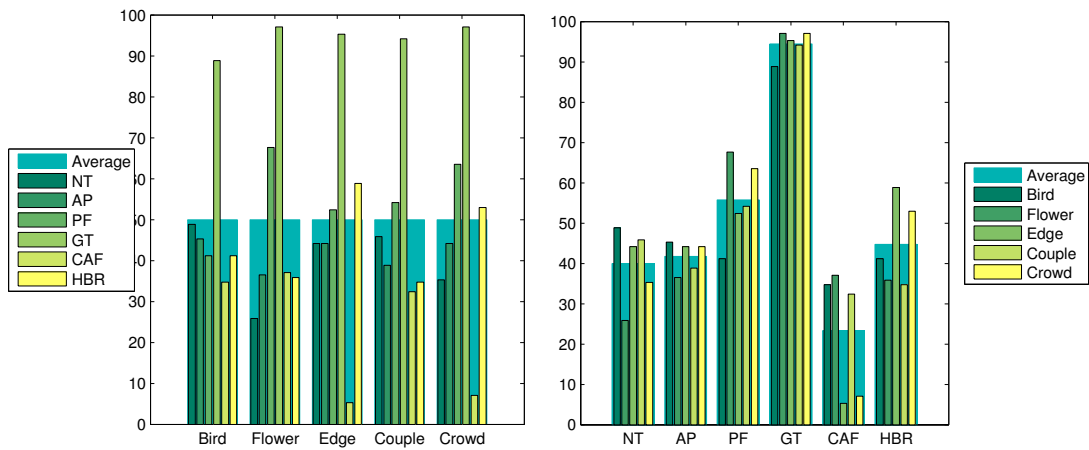
Another interesting finding is the relatively uniform distribution of PC scores for the image *Bird*, and thus the lowest value of GT (77.65). By taking a closer look at the corresponding box plot shown in Figure 4.7(b) for *Bird*, one can see the large fluctuation of PC scores for the individual inpainting approaches. Moreover, the scores of GT also shows the largest deviation for this image (see right box plot in Figure 4.7(b)). The observers declared that they found it hard to detect any differences, which might be due to the fact that *Bird* exhibits the least number of target pixels, and those are additionally located in primarily low textured areas. Hence, the less significant PC scores of *Bird* are discarded in Figure 4.7(c). As can be seen, when only the different inpainting approaches are considered (i.e. without GT values), the proposed methods DPM and APM perform best, yielding a score of 75.55 and 60.29, respectively. Additionally, a detailed comparison of the individual approaches is given in Figure A.1 listed in Appendix A. As can be seen, the proposed depth-based method DPM outperforms CAF at every sequence and even reaches a PC score of 100 for image *Arm* in comparison to HBR.

4.3.2 Video sequences

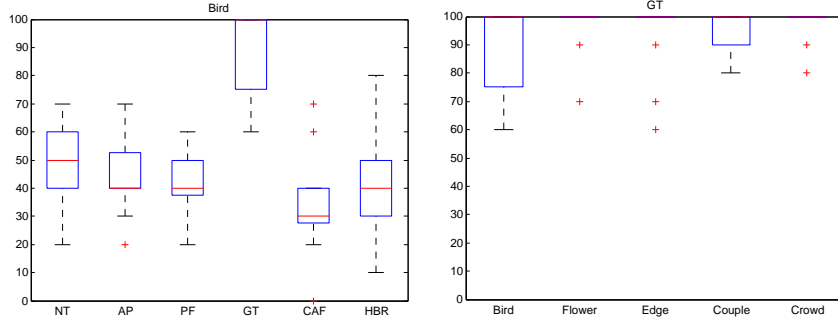
In Figure 4.12, the PC scores of the chosen videos sequences are illustrated. Obviously, compared to still images, the GT data have been identified even more clearly by the observers. Thus, GT reaches a preference ratio of 94.47%, since the influence of warping artifacts comes more into effect for moving pictures. Moreover, as stated before, AP (41.76) only provides a negligible improvement over NT (40.00) due to the rejection of invalid NN correspondences caused by both camera motion and foreground movement.

Similar to the PC scores of still images, HBR also achieves a significantly higher subjective rating compared to the objective quality metrics. In particular, although both inpainting approaches fill in the missing information frame by frame, HBR clearly outperforms CAF. This becomes apparent especially in the highly textured sequences *Edge* and *Crowd*, where HBR yields PC scores of 58.82 and 52.94 as opposed to 5.29 and 7.06, respectively, since flickering artifacts are entirely omitted by the replication of nearby image portions. Nevertheless, the proposed temporal method PF achieves the best ranking among all inpainting approaches, where the maximum scores are received for the video sequences *Flower* (67.65) and *Edge* (63.53).

Furthermore, a significant drop in the rating of GT can be recognized again at the sequence *Bird*. As shown in the box plots in Figure 4.12(b), the median values of all inpainting approaches range from 30 to at most 50, which indicates a great amount of equal valuations (i.e. no specified preference). Additionally, *Bird* exhibits the largest deviation in PC scores for GT among all input sequences. The study participants noted that the assessment of videos is much more difficult in principle, since the user cannot focus on a specific part of the frame and compare the differences by interactively switching between the two input videos, as it is possible for still images. Moreover, as the *Bird* sequence additionally features a lot of movement of the foreground objects as well as increased camera motion, the observers solely identified any adverse differences at the beak of the “Bird”. However, as shown in Figure 4.13, these visual interferences are caused by warping artifacts that are already present in the input data of the image completion systems. Hence, as for still images, Figure 4.12 illustrates the PC scores discarding the less

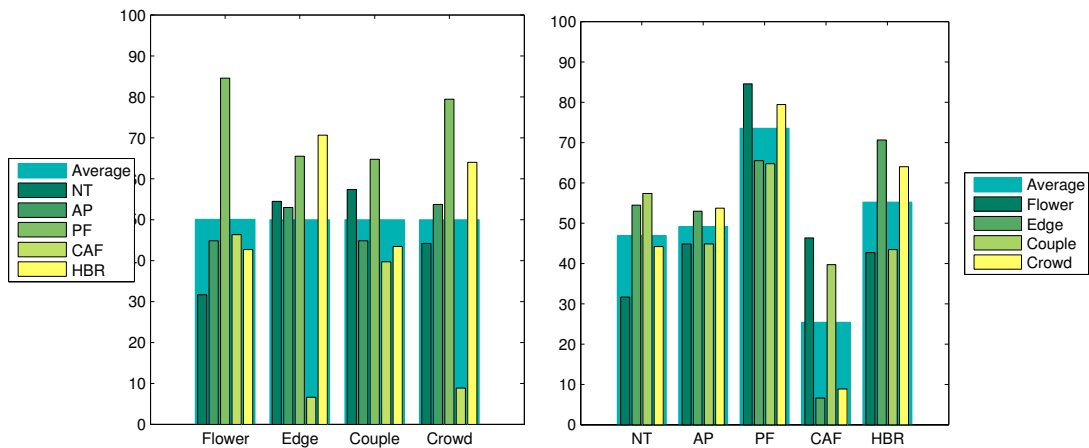


(a) PC scores: per sequence (left) and per method (right).



(b) Box plots for *Bird* (left) and *GT* (right).

On each box, the red line indicates the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered as outliers (red crosses). Points are drawn as outliers if they are larger than $q_3 + 1.5 \cdot (q_3 - q_1)$ or smaller than $q_1 - 1.5 \cdot (q_3 - q_1)$, where q_1 and q_3 are the 25th and 75th percentiles, respectively.



(c) PC scores discarding *Bird* and *GT*.

Figure 4.12: Pair comparison scores for video sequences.

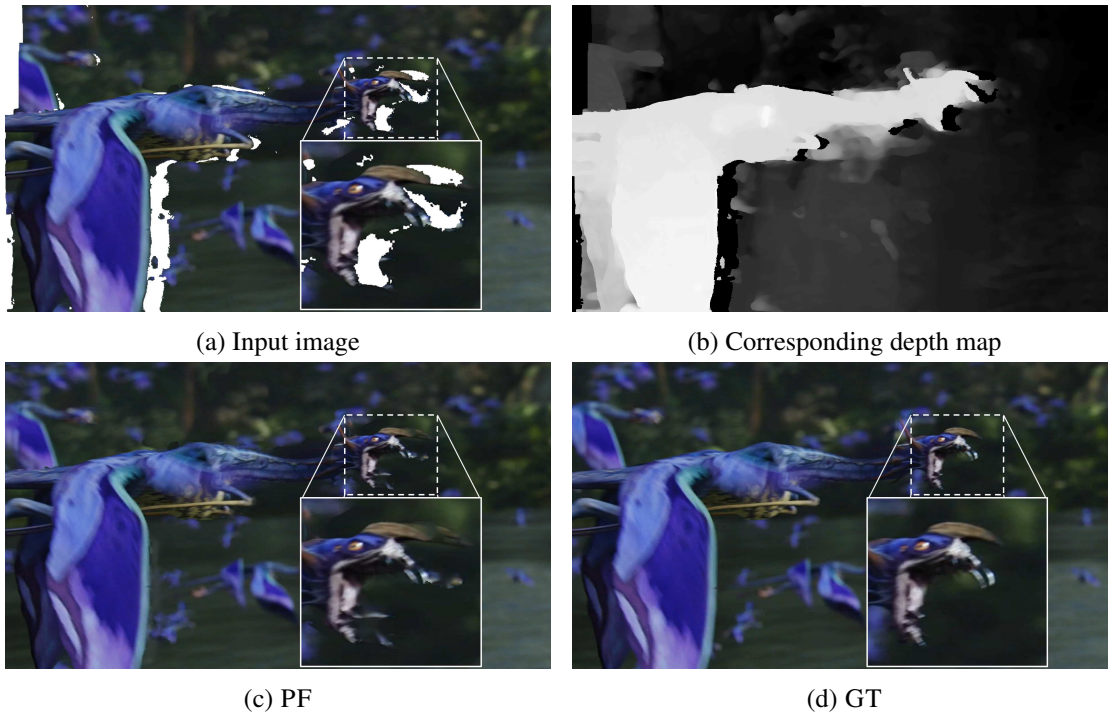


Figure 4.13: Sample frame of *Bird* sequence showing visual disturbances due to warping artifacts.

significant results of sequence *Bird*. It can be seen that the proposed approach PF performs best, yielding an average rate of 73.53. Additionally, a comprehensive comparison of the individual video inpainting approaches can be found in Figure A.2 listed in Appendix A.

4.4 Runtime analysis

Finally, the runtime results of the different image and video completion approaches are briefly addressed. As can be seen in Figure 4.14, the simple inpainting method HBR achieves over-real-time performance of 10 milliseconds per frame at a moderate PC score. Moreover, compared to the professional hole filling approach CAF, the proposed algorithms are up to four magnitudes of order slower but yield significantly better comparison results. However, it should be noted that in the proposed completion framework hardly any runtime optimization has been done yet, since the focus was placed primarily on the enhancement of the inpainting quality. So, there is plenty of room left for further improvements which will be of special interest in future work.

Another interesting finding is that the initial PatchMatch-based approach PM and the proposed adaptation thereof APM have almost identical runtimes but vary significantly in their obtained PC scores. Furthermore, by taking a closer look at Figure 4.15, one can see that the runtime distribution of the individual stages in the image completion pipeline differs distinctly. Although the initial runtime of APM increases due to the additional preprocessing step, the

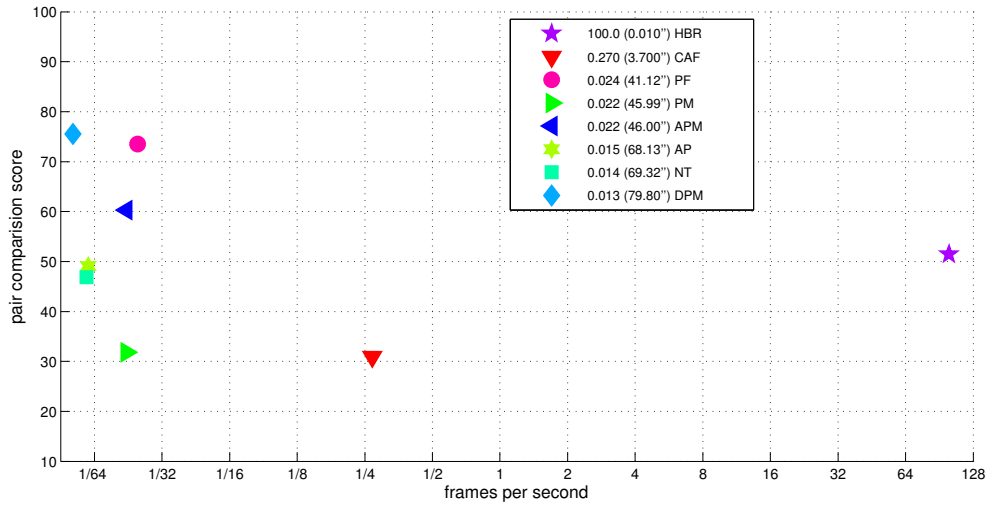


Figure 4.14: Pair comparison score versus runtime of image and video inpainting approaches.

computational expense is reduced in the matching stage by the use of adaptive patch sizes. Particularly, a single matching pass of APM according to an average patch size of 45.8 pixels takes 4.04 seconds (sec) in contrast to 6.98 sec using fixed window sizes of 51 pixels in PM. Thus, the matching stage corresponds to 52.73% and 91.07% of the total runtime of APM and PM, respectively. On the other hand, since the number of valid candidate exemplars decreases in DPM due to the predefined depth constraints, the patch sizes increase to ensure a minimal amount of valid pixels in each target patch according to the specified threshold. Hence, since the calculation of the patch distance metric denotes the most time consuming part in the image completion workflow, DPM has the longest runtime (9.22 sec per matching pass) due to an average patch size of 65.8 pixels.

Furthermore, it is noteworthy that the video inpainting approach AP is 1.19 sec faster than NT on average, since 0.72% fewer correspondence updates are performed in consequence of ANNF propagation. Additionally, the lowest runtime of PF results from the selective use of depth information.

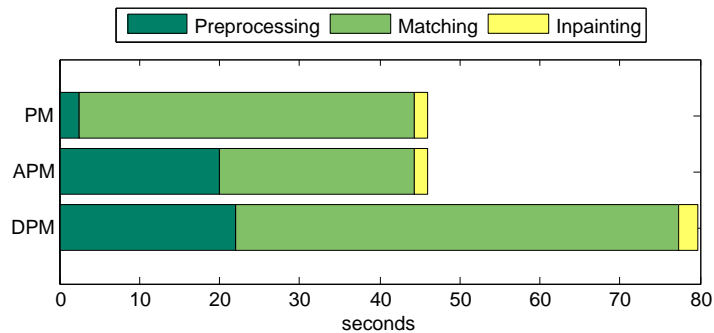


Figure 4.15: Runtime distribution of proposed image inpainting approaches.

Conclusion and Future Work

In this thesis, the problem of filling *disocclusions* in the field of stereoscopic images and 3D videos has been addressed. Disocclusion denotes a common challenge in *Depth Image Based Rendering* (DIBR) where areas that are occluded in the original view become visible in the newly rendered virtual view. To synthesize visually plausible content in the disoccluded regions, a texture-oriented inpainting approach was presented based on the nearest-neighbor search algorithm *PatchMatch* of Barnes et al. [BSF+09].

To overcome problems induced by poor initialization, a so-called *onetime* inpainting approach was proposed that uses only valid (i.e. already present) image information in the completion process as opposed to newly synthesized data. We have demonstrated that the initial *PatchMatch* algorithm produces insufficient inpainting results in regions where the fixed patch sizes are smaller than the hole to be filled, since in those cases no visual information is available for the calculation of the patch distance metric. To account for these non-occupied target patches, adaptive patch sizes in conjunction with a valid matching pixel threshold were introduced to ensure a minimal amount of valid pixels for the determination of patch similarities. This approach also helps to gently propagate information from the target boundary into the hole, as the overlap of neighboring patches close to the hole margin is reduced and, consequently, fewer patches are involved in the calculation of the color value of an individual pixel.

Furthermore, since inaccuracies in the depth estimation process as well as pixel position quantization after the warping step may strongly affect the quality of the resultant novel view, two additional preprocessing steps were introduced. In particular, unilateral morphological dilation of the disocclusion mask was used to decrease the amount of pixels at the edge of a foreground object that are erroneously labeled as background. Additionally, diffusion-based infilling of minor holes was employed to reduce the overall computation time.

Since disocclusions primarily result from the displacement of objects in the foreground, a reasonable approach is to fill the vacant regions with texture obtained from the background. For that purpose, the *PatchMatch* algorithm was extended to incorporate the available depth information in the matching and inpainting stage to provide appropriate patch correspondences. However, depth maps computed by existing stereo matching techniques exhibit inaccuracies,

e.g. due to untextured regions, which require an attenuation of the depth constraints. Thus, instead of imposing restrictions on the patch level, in the presented approach, depth limits were specified for each individual target region based on an outlier detection. Experimental outcomes have shown that this strategy helps to prevent *ghost shadowing artifacts* in the inpainting results caused by color interference of foreground objects.

The proposed image completion approach was also extended to video sequences. Since a video processing algorithm that only works framewise and thus neglects the inter-frame relations may produce inconsistencies (“*flickering*”), two temporal extensions were presented. First, based on the assumption that the content of two consecutive frames will typically not differ significantly, the approximated patch correspondences of the current frame were used to initialize the ANNF of the subsequent one. Although this frame-to-frame propagation saves computation time as the convergence of the matching process is accelerated, the reduction of patch flickering is not satisfying, especially in dynamic scenes where the position of the disoccluded regions alters considerably. Hence, a second approach was investigated where color information of the previously filled image is used in the inpainting process of the following frame. This entails a reduction of flickering as rapid color changes of consecutive frames are suppressed and additionally diminishes the influence of warping artifacts.

Regarding the quality of the completed still images and video sequences, the various enhancement stages of the proposed inpainting approach were compared to *Horizontal Background Replication* (HBR) and Adobe®’s professional image completion function *Content-Aware Fill* (CAF), which is also a PatchMatch-based inpainting technique. The evaluation results obtained from objective quality metrics, including RMSE, PSNR, MSSIM and *Relative Frame Differential Flicker* (RFDF), showed a lack of adequate image and video quality assessment measurements that are able to reflect the behavior of the human visual system. The relatively large variation of the results between the individual test sequences indicates the dependence of these pixelwise error metrics on the underlying image content. Therefore, a supplementary subjective user study was carried out. The evaluation results of the pair comparison test demonstrated the continuous improvement of the proposed inpainting framework and indicated the superior performance of the presented depth-based and temporal extension compared to CAF and the initial PatchMatch algorithm.

However, there is substantial room for further improvements, especially concerning the runtime performance. To achieve acceptable runtime results similar to CAF, the search for patch correspondences has to be accelerated. For that purpose, recent adaptations of the PatchMatch algorithm like *Coherency Sensitive Hashing* [KA11] and *TreeCANN* [OA12] have shown promising results. Additionally, the application of image pyramids coupled with a selective use of color information, e.g. only at the finest resolution and grayscales for the remaining pyramid layers, could lead to a further speedup due to the use of smaller patch sizes and, consequently, to reduced computational costs in the pixelwise calculation of patch similarities.

Furthermore, although Gaussian-weighted pixel synthesis produces appealing inpainting results, it would be interesting to investigate alternative approaches to reduce the blurriness of the completion outcomes and to enhance the preservation of image structures by the use of e.g. gradient information.

Listings of Evaluation Results

This chapter provides the numerical evaluation results of the objective image/video quality assessment and the subjective user study as presented in Chapter 4. Additionally, the pair comparison results between the individual image and video inpainting approaches are shown.

A.1 Objective metrics

	PM	APM	DPM	CAF	HBR
Bird	17.73 23.16 80.88	17.28 23.38 83.88	18.87 22.62 82.52	20.34 21.96 75.89	17.32 23.36 80.92
Flower	24.70 20.28 81.98	19.54 22.31 85.27	18.62 22.73 85.11	25.56 19.98 77.36	24.39 20.39 80.99
Edge	35.66 17.09 57.50	31.90 18.05 58.82	36.82 16.81 58.39	49.57 14.23 46.92	40.13 16.06 51.47
Arm	19.90 22.15 90.38	17.59 23.23 93.18	14.39 24.97 94.04	21.98 21.29 89.33	21.20 21.60 90.47
Crowd	30.31 18.50 54.02	26.87 19.55 58.89	26.95 19.52 59.45	35.28 17.18 49.61	29.44 18.75 57.89
Average	25.66 20.24 72.95	22.64 21.30 76.01	23.13 21.33 75.90	26.50 20.03 72.35	30.54 18.93 67.82

Table A.1: Objective metric scores of still images (*cf.* Figure 4.3). The columns represent RSME, PSNR and MSSIM, respectively.

	NT	AP	PF	CAF	HBR
Bird	16.99 23.88 83.37 1.907	16.97 23.90 83.45 1.889	16.22 24.29 84.59 1.774	18.77 22.95 80.92 1.990	21.01 21.94 76.05 2.098
Flower	24.02 20.65 82.40 2.512	23.21 20.96 82.86 2.153	26.41 19.83 81.70 1.751	24.94 20.38 80.91 2.500	29.72 18.77 75.57 1.994
Edge	28.72 19.19 61.10 2.421	28.44 19.29 61.51 2.100	23.44 21.11 66.55 1.524	31.37 18.50 56.26 2.696	35.19 17.42 51.72 2.432
Couple	19.92 22.33 74.39 2.345	19.69 22.44 74.57 2.144	20.34 22.10 74.41 1.662	25.21 20.27 67.46 2.863	24.56 20.45 65.24 2.218
Crowd	31.84 18.08 54.53 2.969	31.01 18.34 55.60 2.431	24.24 20.66 64.48 1.590	33.60 17.64 52.22 2.837	40.74 15.96 43.74 2.671
Average	24.30 20.83 71.16 2.431	23.87 20.99 71.60 2.143	22.13 21.60 74.35 1.660	26.78 19.95 67.55 2.577	30.24 18.91 62.46 2.282

Table A.2: Objective metric scores of video sequences (*cf.* Figure 4.4). The values represent RSME, PSNR, MSSIM and RFDF.

A.2 Subjective user study

	PM	APM	DPM	GT	CAF	HBR
Bird	35.88	58.82	42.94	77.65	41.76	42.94
Flower	22.94	64.71	47.65	82.35	25.88	56.47
Edge	30.00	51.76	67.65	81.76	10.59	58.24
Arm	36.47	57.65	75.88	95.88	24.71	9.412
Crowd	21.76	30.59	65.88	91.76	38.82	51.18
Average	29.41	52.71	60.00	85.88	28.35	43.65

Table A.3: Pair comparison scores of still images (*cf.* Figure 4.7(a)).

	NT	AP	PF	GT	CAF	HBR
Bird	48.82	45.29	41.18	88.82	34.71	41.18
Flower	25.88	36.47	67.65	97.06	37.06	35.88
Edge	44.12	44.12	52.35	95.29	5.294	58.82
Couple	45.88	38.82	54.12	94.12	32.35	34.71
Crowd	35.29	44.12	63.53	97.06	7.059	52.94
Average	40.00	41.76	55.76	94.47	23.29	44.71

Table A.4: Pair comparison scores of video sequences (*cf.* Figure 4.12(a)).

	PM	APM	DPM	CAF	HBR
Flower	25.74	76.47	53.68	30.88	63.24
Edge	33.09	60.29	76.47	13.24	66.91
Arm	45.59	69.12	92.65	30.88	11.76
Crowd	22.79	35.29	79.41	48.53	63.97
Average	31.80	60.29	75.55	30.88	51.47

Table A.5: Pair comparison scores of still images discarding GT and *Bird* (cf. Figure 4.7(c)).

	NT	AP	PF	CAF	HBR
Flower	31.62	44.85	84.56	46.32	42.65
Edge	54.41	52.94	65.44	6.618	70.59
Couple	57.35	44.85	64.71	39.71	43.38
Crowd	44.12	53.68	79.41	8.824	63.97
Average	46.88	49.08	73.53	25.37	55.15

Table A.6: Pair comparison scores of videos sequences discarding GT and *Bird* (cf. Figure 4.12(c)).

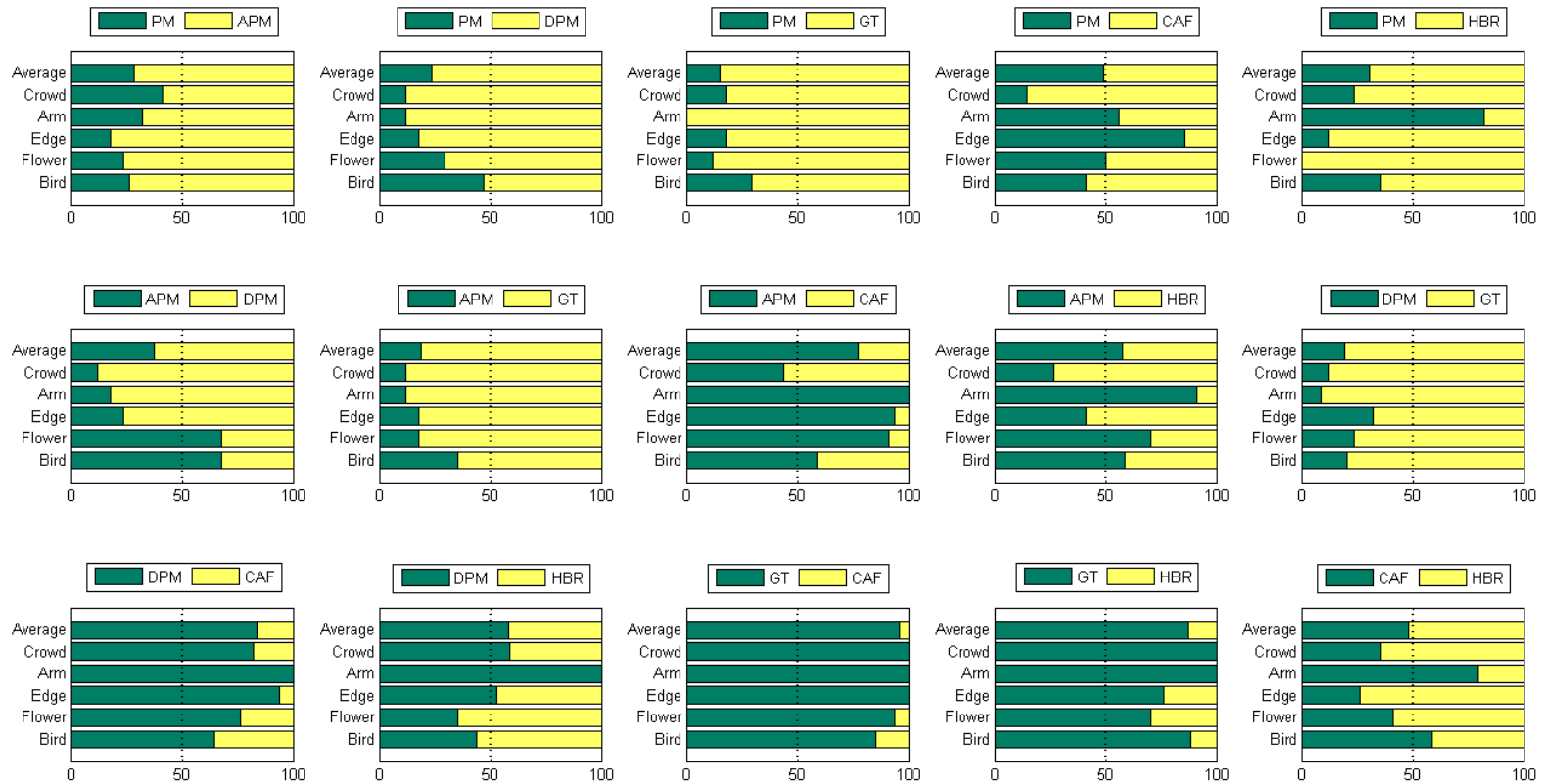


Figure A.1: Pairwise comparison of image inpainting approaches.

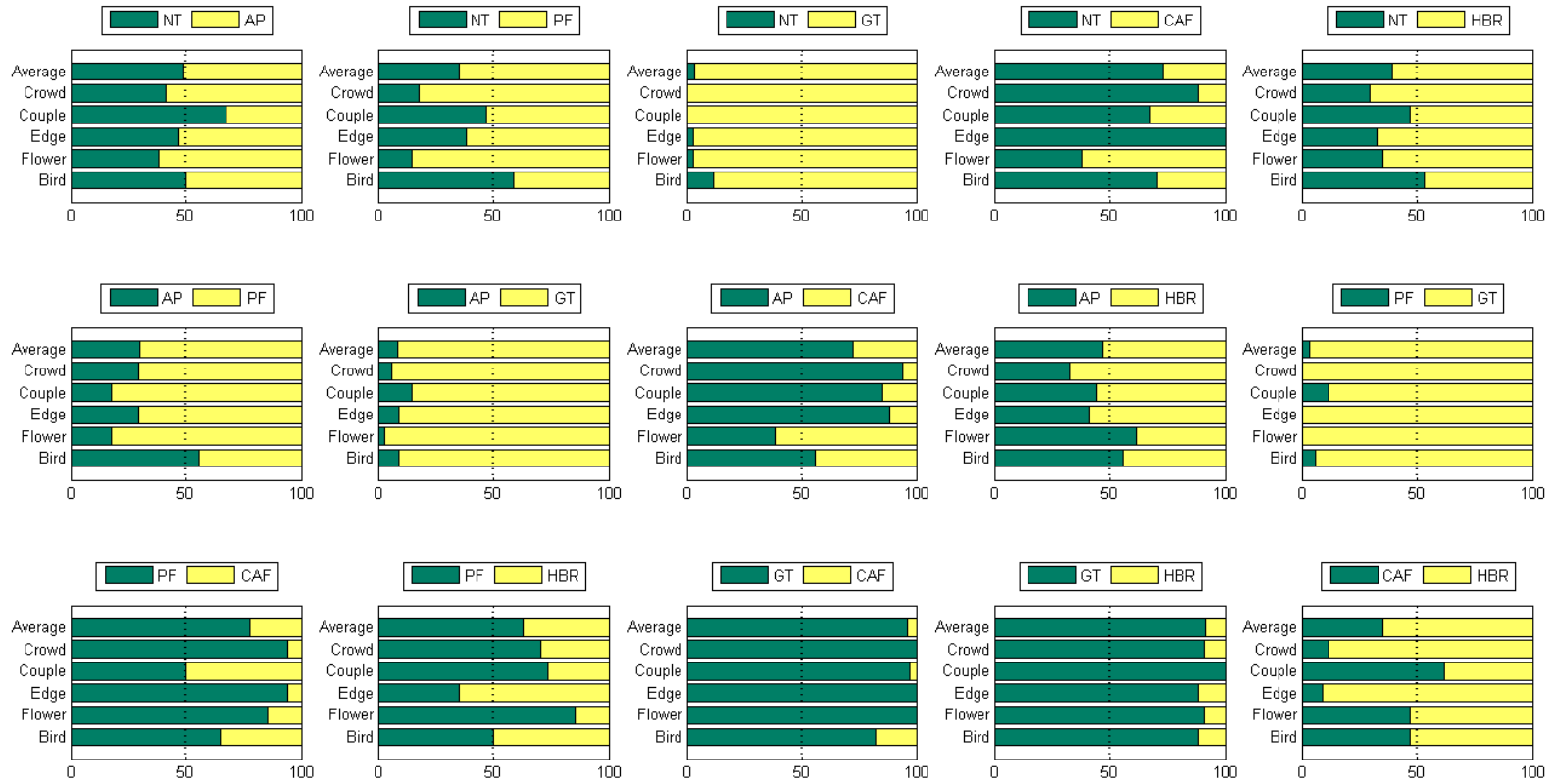


Figure A.2: Pairwise comparison of video inpainting approaches.

Bibliography

- [A01] M. Ashikhmin. “Synthesizing natural textures”. In: *Proceedings of the Symposium on Interactive 3D Graphics (I3D)*, pp. 217–226, Mar. 2001.
- [ABP09] D. Alessandrini, R. Balter, and S. Pateux. “Efficient and automatic stereoscopic videos to N views conversion for autostereoscopic displays”. In: *Proceedings of the SPIE Conference on Stereoscopic Displays and Applications XX*. Vol. 7237, pp. 72371H–12, Feb. 2009.
- [AD13] B. A. Ahire and N. A. Deshpande. “A review on video inpainting techniques”. In: *International Journal of Computer Engineering and Technology (IJCET)* 4.1, pp. 203–210, Jan. 2013.
- [AFC+11] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro. “A variational framework for exemplar-based image inpainting”. In: *International Journal of Computer Vision (IJCV)* 93.3, pp. 319–347, July 2011.
- [AK12] I. Ahn and C. Kim. “Depth-based disocclusion filling for virtual view synthesis”. In: *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 109–114, July 2012.
- [AN12] S. Anitha and B. S. Nagabhushana. “Quality assessment of resultant images after processing”. In: *IISTE Computer Engineering and Intelligent Systems (CEIS)* 3.7, pp. 105–112, July 2012.
- [APS12] A. R. Abraham, A. K. Prabhavathy, and J. D. Shree. “A survey on video inpainting”. In: *International Journal of Computer Applications (IJCA)* 55.9, pp. 43–47, Oct. 2012.
- [AR10] Adobe Research. *Content-aware fill*. <http://www.adobe.com/technology/projects/content-aware-fill.html>. Accessed: 2013-10-21.
- [B00] G. Bradski. “The OpenCV library”. In: *Dr. Dobb’s Journal of Software Tools*, 2000.
- [B11] A. Bogner. “Evaluierung und Entwurf von Epipolarrektifizierungsverfahren zur Verwendung in einem Stereovision Framework”. MA thesis, Institute of Software Technology and Interactive Systems, Vienna University of Technology, 2011.
- [B75] J. L. Bentley. “Multidimensional binary search trees used for associative searching”. In: *Communications of ACM* 18.9, pp. 509–517, Sept. 1975.

- [BBS01] M. Bertalmio, A. L. Bertozzi, and G. Sapiro. “Navier-stokes, fluid dynamics, and image and video inpainting”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1, pp. I:355–I:362, Dec. 2001.
- [BG08] M. Bleyer and M. Gelautz. “Simple but effective tree structures for dynamic programming-based stereo matching”. In: *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, pp. 415–422, Jan. 2008.
- [BSC+00] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. “Image inpainting”. In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 417–424, July 2000.
- [BSF+09] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. “PatchMatch: a randomized correspondence algorithm for structural image editing”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 28.3, pp. 24:1–24:11, Aug. 2009.
- [BSG+10] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. “The generalized PatchMatch correspondence algorithm”. In: *Proceedings of the 11th European Conference on Computer vision: Part III (ECCV)*, pp. 29–43, Sept. 2010.
- [BVS+03] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. “Simultaneous structure and texture image inpainting”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2, pp. II:707–712, June 2003.
- [C11] V. Caselles. “Exemplar-based image inpainting and applications”. In: *International Council for Industrial and Applied Mathematics SIAM News (ICIAM)* 44.10, pp. 1–3, Dec. 2011.
- [CLH12] K.-M. Chang, T.-C. Lin, and Y.-M. Huang. “Parallax-guided disocclusion inpainting for 3D view synthesis”. In: *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE)*, pp. 398–399, Jan. 2012.
- [CLL11] C.-M. Cheng, S.-J. Lin, and S.-H. Lai. “Spatio-temporally consistent novel view synthesis algorithm from video-plus-depth sequences for autostereoscopic displays”. In: *IEEE Transactions on Broadcasting (TBC)* 57.2, pp. 523–532, 2011.
- [CM02] D. Comaniciu and P. Meer. “Mean shift: a robust approach toward feature space analysis”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 24.5, pp. 603–619, May 2002.
- [CPT03] A. Criminisi, P. Perez, and K. Toyama. “Object removal by exemplar-based inpainting”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2, pp. II:721–II:728, June 2003.
- [CS00] T. F. Chan and J. Shen. *Mathematical models for local nontexture inpaintings*. Tech. rep. UCLA CAM TR 00-11, Mar. 2000.
- [CS01] T. F. Chan and J. Shen. “Nontexture inpainting by curvature-driven diffusions (CDD)”. In: *Journal of Visual Communication and Image Representation (JVCI)* 12.4, pp. 436–449, Dec. 2001.

- [CZV06] S. S. Cheung, J. Zhao, and M. V. Venkatesh. “Efficient object-based video inpainting”. In: *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 705–708, Oct. 2006.
- [DCY03] I. Drori, D. Cohen-Or, and H. Yeshurun. “Fragment-based image completion”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH) 22.3*, pp. 303–312, July 2003.
- [DP10] I. Daribo and B. Pesquet-Popescu. “Depth-aided image inpainting for novel view synthesis”. In: *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pp. 167–170, Oct. 2010.
- [EL99] A. A. Efros and T. K. Leung. “Texture synthesis by non-parametric sampling”. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV)*. Vol. 2, pp. 1033–1038, Sept. 1999.
- [ESG12] M. Eisenbarth, F. Seitner, and M. Gelautz. “Quality analysis of virtual views on stereoscopic video content”. In: *Proceedings of the 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 333–336, Apr. 2012.
- [F08] I. B. Fidaner. *A survey on variational image inpainting, texture synthesis and image completion*. <http://issuu.com/fidaner/docs/3012627-a-survey-on-variational-image-inpainting-t>. Accessed: 2013-10-30.
- [GS11] A. Ghanbari and M. Soryani. “Contour-based video inpainting”. In: *Proceedings of the 7th Iranian Conference on Machine Vision and Image Processing (MVIP)*, pp. 1–5, Nov. 2011.
- [HB10] J. Herling and W. Broll. “Advanced self-contained object removal for realizing real-time diminished reality in unconstrained environments”. In: *Proceedings of the 9th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 207–212, Oct. 2010.
- [HBG11] L. He, M. Bleyer, and M. Gelautz. “Object removal by depth-guided inpainting”. In: *Austrian Association for Pattern Recognition Workshop (ÖAGM / AAPR)*, pp. 1–8, May 2011.
- [HHA12] Y. Hwang, B. Han, and H.-K. Ahn. “A fast nearest neighbor search algorithm by nonlinear embedding”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3053–3060, June 2012.
- [HS12] K. He and J. Sun. “Statistics of patch offsets for image completion”. In: *Proceedings of the 12th European conference on Computer Vision: Part II (ECCV)*, pp. 16–29, Florence, Italy, Oct. 2012.
- [HWD+08] C. T. Hewage, S. T. Worrall, S. Dogan, and A. M. Kondoz. “Prediction of stereoscopic video quality using objective quality models of 2-D video”. In: *Electronics Letters (EL) 44.16*, pp. 963–965, Aug. 2008.
- [IM98] P. Indyk and R. Motwani. “Approximate nearest neighbors: towards removing the curse of dimensionality”. In: *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 604–613, Dallas, Texas, USA, May 1998.

- [ITU+12] ITU-R Recommendation BT.2021. *Subjective methods for the assessment of stereoscopic 3DTV systems*. Tech. rep. International Telecommunication Union, Mar. 2012.
- [JJ12] V. Janarthanan and G. Jananii. “A detailed survey on various image inpainting techniques”. In: *Bonfring International Journal of Advances in Image Processing (BIJAIP)* 2.3, pp. 1–3, Sept. 2012.
- [JT03] J. Jia and C.-K. Tang. “Image repairing: robust image synthesis by adaptive ND tensor voting”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1, pp. I:643–I:650, June 2003.
- [JTW06] J. Jia, Y.-W. Tai, T.-P. Wu, and C.-K. Tang. “Video repairing under variable illumination using cyclic motions”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 28.5, pp. 832–839, May 2006.
- [K79] G. Kanizsa. *Organization in vision: essays on gestalt perception*. Praeger, 1979.
- [KA11] S. Korman and S. Avidan. “Coherency sensitive hashing”. In: *Proceedings of the 2011 International Conference on Computer Vision (ICCV)*, pp. 1607–1614, Nov. 2011.
- [KBB+05] S. Kumar., M. Biswas, S. J. Belongie, and T. Q. Nguyen. “Spatio-temporal texture synthesis and image inpainting for video applications”. In: *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. Vol. 2, pp. II:85–II:88, Sept. 2005.
- [KK03] J. B. Kim and H. J. Kim. “Region removal and restoration using a genetic algorithm with isophote constraint”. In: *Pattern Recognition Letters (PRL)* 24.9–10, pp. 1303–1316, June 2003.
- [KND+10] M. Köppel, P. Ndjiki-Nya, D. Doshkov, H. Lakshman, P. Merkle, K. Müller, and T. Wiegand. “Temporally consistent handling of disocclusions with texture synthesis for depth-image-based rendering”. In: *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP)*, pp. 1809–1812, Sept. 2010.
- [KP92] J. M. Kasson and W. Plouffe. “An analysis of selected computer interchange color spaces”. In: *ACM Transactions on Graphics (TOG)* 11.4, pp. 373–405, Oct. 1992.
- [KR02] D. R. Karger and M. Ruhl. “Finding nearest neighbors in growth-restricted metrics”. In: *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 741–750, May 2002.
- [KWD+12] M. Köppel, X. Wang, D. Doshkov, T. Wiegand, and P. Ndjiki-Nya. “Consistent spatio-temporal filling of disocclusions in the multiview-video-plus-depth format”. In: *Proceedings of the 14th IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pp. 25–30, Sept. 2012.
- [LLL+11] C.-H. Ling, Y.-M. Liang, C.-W. Lin, Y.-S. Chen, and H.-Y. M. Liao. “Human object inpainting using manifold learning-based posture sequence estimation.” In: *IEEE Transactions on Image Processing (TIP)* 20.11, pp. 3124–3135, Nov. 2011.

- [LLL+12] Y. Lai, X. Lan, Y. Liu, and N. Zheng. “Disocclusion using depth reliability map for view synthesis”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1449–1452, Mar. 2012.
- [M10] V. V. Mahalingam. “Digital inpainting algorithms and evaluation”. PhD thesis, University of Kentucky, Apr. 2010.
- [MHC+12] B. Morse, J. Howard, S. Cohen, and B. Price. “PatchMatch-based content completion of stereo image pairs”. In: *Proceedings of the Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pp. 555–562, June 2012.
- [MOS13] S. M. Muddala, R. Olsson, and M. Sjöström. “Disocclusion handling using depth-based inpainting”. In: *Proceedings of the Fifth International Conferences on Advances in Multimedia (MMEDIA)*, pp. 1–6, Apr. 2013.
- [MSD+08] K. Müller, A. Smolic, K. Dix, P. Kauff, and T. Wiegand. “Reliability-based generation and view synthesis in layered depth video”. In: *Proceedings of the 10th IEEE Workshop on Multimedia Signal Processing (MMSP)*, pp. 34–39, Dec. 2008.
- [MV12] K. S. Mahajan and M. B. Vaidya. “Image in painting techniques: a survey”. In: *IOSR Journal of Computer Engineering (IOSRJCE)* 5.4, pp. 45–49, Sept. 2012.
- [NKD+11] P. Ndjiki-Nya, M. Köppel, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, and T. Wiegand. “Depth image based rendering with advanced texture synthesis for 3-D video”. In: *IEEE Transactions on Multimedia (TMM)* 13.3, pp. 453–465, June 2011.
- [OA12] I. Olonetsky and S. Avidan. “TreeCANN - k-d tree coherence approximate nearest neighbor algorithm”. In: *Proceedings of the 12th European Conference on Computer Vision: Part IV (ECCV)*, pp. 602–615, Florence, Italy, Oct. 2012.
- [OBM+01] M. M. Oliveira, B. Bowen, R. McKenna, and Y.-S. Chang. “Fast digital image inpainting”. In: *Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP)*, pp. 261–266, 2001.
- [PM90] P. Perona and J. Malik. “Scale-space and edge detection using anisotropic diffusion”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 12.7, pp. 629–639, July 1990.
- [PSB05] K. A. Patwardhan, G. Sapiro, and M. Bertalmio. “Video inpainting of occluding and occluded objects”. In: *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. Vol. 2, pp. II:69–II:72, Sept. 2005.
- [RB12] S. A. Ramakanth and R. V. Babu. “Feature match: an efficient low dimensional PatchMatch technique”. In: *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, pp. pages, Dec. 2012.
- [RP66] A. Rosenfeld and J. L. Pfaltz. “Sequential operations in digital picture processing”. In: *Journal of the Association for Computing Machinery (JACM)* 13.4, pp. 471–494, Oct. 1966.

- [RPK12] F. Raimbault, F. Pitie, and A. Kokaram. “Stereo video completion for rig and artefact removal”. In: *Proceedings of the 13th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pp. 1–4, May 2012.
- [RPM+13] S. Ravi, P. Pasupathi, S. Muthukumar, and N. Krishnan. “Image in-painting techniques - a survey and analysis”. In: *Proceedings of the 9th International Conference on Innovations in Information Technology (IIT)*, pp. 36–41, Mar. 2013.
- [S95] J. A. Sethian. “A fast marching level set method for monotonically advancing fronts”. In: *Proceedings of the National Academy of Sciences*. Vol. 93. 4, pp. 1591–1595, Oct. 1995.
- [SA12] M. Solh and G. AlRegib. “Hierarchical hole-filling for depth-based view synthesis in FTV and 3D video”. In: *IEEE Journal of Selected Topics in Signal Processing (J-STSP)* 6.5, pp. 495–504, Sept. 2012.
- [SC05] T. K. Shih and R.-C. Chang. “Digital inpainting - survey and multilayer image inpainting algorithms”. In: *Proceedings of the Third International Conference on Information Technology and Applications (ICITA)*. Vol. 1, pp. 15–24, July 2005.
- [SJ10] M. Schmeing and X. Jiang. “Depth image based rendering: a faithful approach for the disocclusion problem”. In: *Proceedings of the 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1–4, May 2010.
- [SJ11] M. Schmeing and X. Jiang. “Time-consistency of disocclusion filling algorithms in depth image based rendering”. In: *Proceedings of the 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1–4, May 2011.
- [SKK+11] A. Smolic, P. Kauff, S. Knorr, A. Hornung, M. Kunter, M. Muller, and M. Lang. “Three-dimensional video postproduction and processing”. In: *Proceedings of the IEEE (PROC)* 99.4, pp. 607–625, Apr. 2011.
- [SLW+03] T. K. Shih, L.-C. Lu, Y.-H. Wang, and R.-C. Chang. “Multi-resolution image inpainting”. In: *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*. Vol. 2, pp. 485–488, July 2003.
- [SMD+08] A. Smolic, K. Müller, K. Dix, P. Merkle, P. Kauff, and T. Wiegand. “Intermediate view interpolation based on multiview video plus depth for advanced 3D video systems”. In: *Proceedings of the 15th IEEE International Conference on Image Processing (ICIP)*, pp. 2448–2451, Oct. 2008.
- [SN12] H. Sureka and P. J. Narayanan. “Mixed-resolution patch-matching”. In: *Proceedings of the 12th European conference on Computer Vision: Volume Part VI (ECCV)*, pp. 187–198, Florence, Italy, Oct. 2012.
- [SP07] D. Scharstein and C. Pal. “Learning conditional random fields for stereo”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, June 2007.

- [SS02] D. Scharstein and R. Szeliski. “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms”. In: *International Journal of Computer Vision (IJCV)* 47.1-3, pp. 7–42, Apr. 2002.
- [SS03] D. Scharstein and R. Szeliski. “High-accuracy stereo depth maps using structured light”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1, pp. I:195–I:202, June 2003.
- [SSB11] K. Sangeetha, P. Sengottuvelan, and E. Balamurugan. “Comparative analysis and evaluation of image imprinting algorithms”. In: *Journal of Information Engineering and Applications (JIEA)* 1.5, pp. 506–517, May 2011.
- [T04] A. Telea. “An image inpainting technique based on the fast marching method”. In: *Journal of Graphics Tools (JGT)* 9.1, pp. 23–34, 2004.
- [TLD07] Z. Tauber, Z.-N. Li, and M. S. Drew. “Review and preview: disocclusion by inpainting for image-based rendering”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews (TSMCC)* 37.4, pp. 527–540, 2007.
- [TZL02] X. Tong, J. Zhang, L. Liu, X. Wang, B. Guo, and H.-Y. Shum. “Synthesis of bidirectional texture functions on arbitrary surfaces”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 21.3, pp. 665–672, July 2002.
- [VJ01] P. Viola and M. Jones. “Rapid object detection using a boosted cascade of simple features”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1, pp. I:511–I:518, Dec. 2001.
- [VTS06] C. Vázquez, W. J. Tam, and F. Speranza. “Stereoscopic imaging: filling disoccluded areas in depth image-based rendering”. In: *Proceedings of the SPIE Conference on Three-Dimensional TV, Video and Display*. Vol. 6392, pp. 63920D–12, Oct. 2006.
- [WAD+97] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. “Pfinder: real-time tracking of the human body”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.7, pp. 780–785, July 1997.
- [WBB+13] K. Wang, K. Brunnström, M. Barkowsky, M. Urvoy, M. Sjöström, P. Le Callet, S. Tourancheau, and B. André. “Stereoscopic 3D video coding quality evaluation with 2D objective metrics”. In: *Proceedings of the SPIE Conference on Electronic Imaging*. Vol. 8648, pp. 86481L–7, Feb. 2013.
- [WBS+04] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Transactions on Image Processing (TIP)* 13.4, pp. 600–612, Apr. 2004.
- [WJY+08] L. Wang, H. Jin, R. Yang, and M. Gong. “Stereoscopic inpainting: joint color and depth completion from stereo images”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, June 2008.
- [WL00] L.-Y. Wei and M. Levoy. “Fast texture synthesis using tree-structured vector quantization”. In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 479–488, July 2000.

- [WSI07] Y. Wexler, E. Shechtman, and M. Irani. “Space-time completion of video”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 29.3, pp. 463–476, Mar. 2007.
- [WYY+09] X. Wang, M. Yu, Y. Yang, and G. Jiang. “Research on subjective stereoscopic image quality assessment”. In: *Proceedings of the SPIE Conference on Multimedia Content Access: Algorithms and Systems III*. Vol. 7255, pp. 725509–10, Jan. 2009.
- [XS10] Z. Xu and J. Sun. “Image inpainting by patch propagation using patch sparsity”. In: *IEEE Transactions on Image Processing (TIP)* 19.5, pp. 1153–1165, Dec. 2010.
- [ZT05] L. Zhang and W. J. Tam. “Stereoscopic image generation based on depth images for 3D TV”. In: *IEEE Transactions on Broadcasting (TBC)* 51.2, pp. 191–199, 2005.
- [ZZ10] H. Zhou and J. Zheng. “Adaptive patch size determination for patch-based image completion”. In: *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP)*, pp. 421–424, Sept. 2010.