DISS. ETH NO. 22257

Advanced Energy Optimization Algorithms for 2D and 3D Video-Based Graphics Applications

A thesis submitted to attain the degree of
DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by
Manuel Johannes Lang
MSc ETH CS, ETH Zurich, Switzerland
born on 16.10.1982
citizen of Austria

accepted on the recommendation of
Prof. Dr. Markus Gross, examiner
Dr. Aljoša Smolić, co-examiner
Prof. Dr. Marc Pollefeys, co-examiner

2014

Abstract

Videos have become an important part in our daily lives. Thankfully, the recent shift to an entire digital video pipeline allowed the introduction of improved tools and algorithms for professional video processing. Therefore, despite its complexity, video processing has never been easier to use and more available to the general public than it is today.

However, due to the rapid technological advances new challenges in video processing have emerged. The ability to capture high definition content has often outpaced the progress of computing hardware. Furthermore, with the renaissance of stereoscopic 3D a new dimension is added to video processing algorithms. This additionally increases computational demands as well as it introduces its own sets of problems. Moreover, the ubiquitous of currently available displays results in a wide range of possible video output formats. Viewers consume video content on small handhelds as well as on high definition glasses-free 3D displays. This requires sophisticated video processing methods to ensure best viewing conditions on all output devices.

This thesis aims at improving the way 2D and 3D video content can be adapted and artistically enhanced. The first two challenges emerge from the content-display gap. Often the resolution and aspect ratio of the output display may significantly vary from the original content. This thesis introduces a novel image domain warping (IDW) framework and shows that it can be utilized to fit content to different display dimensions without introducing too much noticeable artifacts. We present a novel efficient system that gives complete artistic control over this retargeting process.

A similar content-display gap also exists for stereoscopic productions. Different display dimensions result in different depth perception. To counteract those undesired depth changes we will extend the technique of IDW to be able to change the depth perception of 3D content. Our novel method does not only allow to close the 3D-content display gap, but also gives artists a new innovative post processing tool to locally modify the depth composition. We introduce disparity mapping operators. These operators formalize insights from perception and production rules and are the basis for a general framework for stereoscopic disparity editing. Moreover, we use the same novel IDW approach to generate multi-view video

from stereoscopic input. This is an essential tool to be able to consume stereoscopic content on advanced glasses-free auto-stereoscopic displays.

One other energy optimization algorithm presented in this thesis tackles the problem of information propagation spatially and temporally in a video sequence. To ensure consistent and content-aware propagation such algorithms usually require complex dependencies and therefore result in a heavy and difficult optimization problems. The general processing framework, which this thesis introduces however, is optimized to still work on high definition content. Due to the reduction of per-pixel complexity and the increase of the amount of information that can be considered simultaneously it solves efficiently and accurately many data propagation problems. We show that our novel video processing framework can significantly improve and accelerate various video-processing tasks including such important tasks as optical flow computation, disparity estimation, depth up sampling, colorization, and saliency computation.

Zusammenfassung

Der Konsum von Videoinhalten wurde in den letzten Jahren zu einem wichtigen Teil unseres täglichen Lebens. Die jüngsten Entwicklungen hin zu einer vollkommen digitalen Video-Pipeline erlaubten es bessere Werkzeuge und Algorithmen für die professionelle Videoverarbeitung einzuführen. Dadurch ist die Videoverarbeitung, trotz ihrer Komplexität, einfacher und weiter verbreitet als jemals zuvor.

Aufgrund des rasanten technologischen Fortschritts sind neue Herausforderungen in der Videoverarbeitung entstanden. Die Fähigkeit immer bessere, hochaufgelöste Inhalte aufzuzeichnen überholt oft die Weiterentwicklung der zur Verarbeitung verwendeten Computer-Hardware. Zusätzlich, mit der Renaissance des 3D-Films, wurde noch eine weitere Verarbeitungsdimension zu den meisten Videoalgorithmen hinzugefügt. Dies erhöht zusätzlich den Rechenaufwand und erzeugt weiters eine ganze Reihe an speziellen neuen Problemen. Des Weiteren ergeben sich auch durch die weite Verbreitung von Displays mit verschiedensten Auflösungen und Ausgabeformaten neue Herausforderungen für die Videoverarbeitung. Zuschauer konsumieren Videoinhalte sowohl auf kleinen Handhelds als wie auch auf grossen High-Definition 3D-Displays. Um trotzdem auf allen Ausgabegeräten eine optimale Qualität zu erhalten, werden hochentwickelte Anpassungsalgorithmen benötigt.

Diese Arbeit befasst sich mit der Verbesserung der Art und Weise wie 2D- und 3D-Videoinhalte technisch sowie auch künstlerisch angepasst werden können. Als Erstes widmen wir uns dem Problem der Anspassung von vorgegebenen 2D Videos an bestimmte Ausgabegeräte (Content-Display-Gap). Oft unterscheidet sich die Auflösung und das Seitenverhältnis des Ausgabegerätes stark von den Eigenschaften der Aufnahme. Um dieses Problem zu lösen stellt diese Arbeit zuerst ein neuartiges Werkzeug zur Video-Verformung im Bildbereich (image domain warping, IDW) vor. Wir zeigen dann, dass mit diesem IDW-Werkzeug Videos an beliebige Seitenverhältnisse angepasst werden können, ohne dass dabei starke Verzerrungen in wichtigen Bereichen der Videos entstehen. Unser neuartiges und effizientes System gewährt die volle Kontrolle über diese Bildverformung und erlaubt dadurch auch die gewollte künstlerische Originalbildkomposition zu erhalten.

Ein ähnliches Video-Anpassungsproblem gibt es auch im Bereich der stereosko-

pischen 3D Videoproduktion. Verschiedene Anzeigegeräte führen hier oft dazu, dass ein und derselbe Inhalt eine andere Tiefenwahrnehmung erzeugt. Um dieser unerwünschten Tiefenänderungen entgegenzuwirken erweitern wir unser Bildverformungswerkzeug so, dass wir die Tiefenwahrnehmung von beliebigen 3D-Inhalten anpassen können. Dieses neue innovative 3D Bearbeitungswerkzeug ermöglicht nicht nur Inhalte an 3D Ausgabegeräte anzupassen, sondern erlaubt es Videobearbeitern und Videokünstlern auch in der Videonachbearbeitung die Tiefenzusammensetzung einer aufgezeichnete Szene zu ändern. Dazu stellen wir das Konzept von Deviationsanpassungsfunktionen vor. Diese Funktionen formalisieren Erkenntnisse aus der Wahrnehmungslehre sowie allgemeine bekannte Produktionsregeln für 3D Inhalte, und sind damit die Grundlage für einen allgemeinen Ansatz zur nachträglichen Bearbeitung der Deviationen und der damit verbundenen Tiefenzusammensetzung von 3D Videos. Wir zeigen auch dass unser auf reiner zweidimensionaler Videoverformung basierendes 3D Bearbeitungswerkzeug zudem dafür verwendet werden kann zusätzliche neue Ansichten aus anderen Blickwinkeln zu erzeugen. Damit ist es ein wichtiges Instrument um stereoskopische Aufnahmen bestehend aus nur zwei Ansichten an Ausgabegeräte mit vielen Ansichten (autostereoskopische Bildschirme) anzupassen.

Ein weiterer Energieoptimierungs-Algorithmus, der in dieser Arbeit beschrieben wird, befasst sich mit dem Problem der räumlichen und zeitlichen Diffusion von Information in einer Videosequenz. Um sicherzustellen dass Information konsistent und unter Berücksichtigung der Videoinhalte propagiert werden können werden oft Algorithmen mit komplexen Abhängigkeiten zwischen allen Bildern und allen Bildpunkten benutzt. Dies führt zu schwierigen und rechenaufwendigen Optimierungsproblemen. In dieser Arbeit führen wir deshalb eine neuartige Videoverarbeitungsmethode ein die auch noch für hochaufgelöste Videos effizient arbeitet. Indem wir den Aufwand pro Bildpunkt reduzieren und damit erreichen dass wir gleichzeitig mehr Bildinformation berücksichtigen können, gelingt es uns viele Informationsdiffusionsproblem effizient und in hoher Qualität zu lösen. Wir zeigen dass unsere neue Methode für viele wichtige Anwendungen wie zum Beispiel die optische Flussberechnung, die Deviationsschätzung, die Tiefenkartenrekonstruktion sowie die Filmkolorierung effizient verwendet werden kann.

Acknowledgments

First of all I would like to express my sincere gratitude to my advisor Prof. Markus Gross. His great lectures affirmed me that working and researching in the field of computer graphics is what I wanted to pursue. Not only was he always able to guide my research with his professional insights, but he also created and managed an unbelievably inspiring, friendly, and fun work environment at ETH Computer Graphics Laboratory and Disney Research Zurich.

I would also like to thank my mentors Aljoša Smolić, Oliver Wang, and Alexander Sorkine-Hornung. Without their continued support this work would not have been possible. I like to thank them for all the fruitful discussions, the great creativity and guidance for finding and solving research problems, as well as their hard work during all of our deadlines.

Many thanks go to all my collaborators, especially Nikolce Stefanoski, Tunc Ozan Aydin, Steven Poulakos, Philipp Krähenbühl, Pierre Greisen, and Simon Heinzle. It was a pleasure to work with them and I am thankful for their effort and commitment to the realized projects. Furthermore, I also want to thank all my friends and co-workers at CGL, IGL, CVG and DRZ for all the help and great fun times we had together not only during working hours.

Finally, I am deeply grateful for the continuous support I received from my friends, roommates and family, especially my parents who supported me in so many ways during my studies.

Contents

CHAPTER 1

Introduction

Videos, movies and other visual media have become an important part in our daily lives. With advances in fields like movie capture, data transmission and display technology, handling of video content seems as easy as delivering printed text or emitting a radio program. However, creating convincing and enjoyable videos requires much more than plain content capture. It is often essential to extensively modify and adapt recorded content.

Such modifications include simple operations like editing, color correction or reframing, but often much more sophisticated methods to completely alter video content are used or entirely new artificial sceneries are created without any capturing. Simple modifications are used on a daily basis in broadcast, whereas professional movie productions extensively use any available tool. As a consequence, nearly none of the visual content shown and consumed today is unaltered. Accordingly, content creation for video is still much more labor and time consuming than for any other media. Thankfully, the recent shift to an entire digital video pipeline allowed the introduction of improved tools and algorithms for video processing. Therefore, despite its complexity, video processing has never been easier to use and more available to the general public than it is today.

Due to the rapid technological advances new challenges in video processing have emerged. The ability to capture high definition content (temporally and spatially) has often outpaced the progress of computing and memory hardware. Furthermore, with the renaissance of stereoscopic 3D content a

new dimension is added to video processing algorithms. This additionally increases computational demands as well as it introduces its own sets of new problems and additional requirements for existing algorithms (e.g., depth awareness, interview consistency). Moreover, when it comes to displaying video content, recent years have brought us a zoo of different displays with diverse aspect ratios, resolutions and other capabilities. Viewers consume media on small unconventionally shaped handheld displays as well as on high definition 4k TVs or glasses-free multi-view 3D screens. This requires sophisticated content creation, transmission and adaptation to ensure best viewing conditions on all output devices.

This thesis aims at improving the way high-definition 2D and 3D video content can be adapted and artistically enhanced. This thesis will introduce and propose a solution to three related digital video post-processing tasks. The first two challenges emerge from the content-display gap. Often, produced content is not consumed on a single type of display. The resolution and aspect ratio of the output display may significantly vary from the original content. This can result in substantial distortions or the necessity for large content excision. This thesis introduces a novel image domain warping (IDW) framework and shows that it can be utilized to fit captured content to different display dimensions without introducing too much noticeable artifacts or the need to cut off content. The adaption of content to different output devices is called video retargeting and aims at hiding indispensable distortions in visually unimportant areas. In its core the proposed IDW framework is a specifically tailored large non-linear energy optimization task. In our work we show how to efficiently construct the energy optimization problem from the video content and the desired retargeting constraints and then present an efficient iterative solver to compute the IDW result in near real time even for high resolution content.

A similar content-display gap also exists for stereoscopic 3D productions. Different display dimensions result in different depth perception for the same content. To counteract those undesired depth changes we will extend the technique of IDW to be able to change the depth perception of captured stereoscopic content. Our novel method does not only allow to close the 3D-content display gap, but also gives artist a new innovative post processing tool to locally modify the stereoscopic 3D depth composition after content capturing.

In the foreseeable future, when autostereoscopic displays are widely available, the content-display gap for 3D content will be even more severe. These glasses-free displays require significantly more than two input views. However, nowadays content is mainly produced at most with two cameras. In

this thesis we will show how to generate additional approximated views from just two input views using our IDW framework. Thereby, we present a practical method that allows consuming stereoscopic 2-view content on autostereoscopic glasses-free displays.

The third main energy optimization technique presented in this thesis tackles the problem of information propagation spatially and temporally in a video sequence. To ensure consistent and content-aware propagation such algorithms usually require complex pixel and frame inter-dependencies and therefore result in a complex optimization problems. This is often prohibitively computationally expensive, especially for high definition content. The general processing framework, which this thesis introduces, is optimized to work on high definition content. Due to the reduction of per-pixel complexity and the increase of the amount of information that can be considered simultaneously (number of frames, pixel neighborhood) it solves efficiently and accurately many data regularization, filtering and data-upsampling problems. We show that our novel video processing framework can significantly improve and accelerate various video-processing task including such important task as optical flow computation, disparity estimation, depth up sampling, colorization, and saliency computation.

Next, we individually introduce in more detail the topics of this thesis. We then continue by summarizing the principal contributions of this work. We finish this introduction with a short outline of the thesis, as well as listing the authors relevant peer-reviewed publications.

1.1 Video Retargeting

Motion picture and video are traditionally produced for a specific target platform such as cinema or TV. In recent years, however, we witness an increasing demand for displaying video content on devices with considerably differing display formats. User studies [Set+05; Kno+07] have shown that for novel formats like smart phones or MP3 players, naive linear downscaling is inappropriate because of the introduced distortions; these platforms require content-aware modification of the video for a comfortable viewing experience. Lately, sophisticated solutions have been proposed which compute feature preserving, non-linear rescaling to the desired target resolution [WGC07; RSA08; WS08]. But despite their very promising results, these techniques focus on particular technical elements and lack the systemic view required for practical video content production and viewing.

This thesis presents a novel, comprehensive framework for video retarget-

ing. The framework combines automatic content-analysis with interactive tools. Within an interactive workflow, the content producer defines global constraints to guide the retargeting process based on the concept of key frame editing. This enables her to annotate video with additional information about the desired scene composition or object saliency which would otherwise be impossible to capture by currently available fully automatic techniques. This process augments the original video format with sparse annotations that are time-stamped and stored with the key frames. During playback our system computes an optimized image domain warp considering both automatically computed constraints as well as the ones defined by annotations. This approach enables us to guarantee a consistent, art directed viewing experience, which preserves important cinematographic or artistic intentions to a maximum extent possible when streaming video to arbitrary output devices.

The most distinctive technical feature of our method compared to previous work is a per-pixel image domain warp to the target resolution. We compute and render it in near real-time using a GPU-based multi-scale solver combined with a novel 2D variant of EWA splatting [Zwi+02]. The pixel-level operations have major benefits over previous methods. For the first time, spatio-temporal constraints can be defined at pixel-accuracy without sacrificing performance. This allows for novel automatic warp constraints, for example, a per-pixel constraint that ensures bilateral temporal coherence and is sensitive to scene cuts. Others constraints retain the sharpness of prevalent object edges without introducing blurring or aliasing into the output video. With our IDW framework we do not require strong global smoothness priors in order to keep the warp field consistent at the pixel level. It thus utilizes the available degrees of freedom more effectively and improves the automatic part of feature preservation.

A further important novelty of the presented method is its elegant conceptual approach for antialiasing. If not properly handled, aliasing arises from the resampling step involved in the retargeting as well as from the alterations of the video signals spectral energy distribution during image domain warping. In this thesis we introduce EWA splatting for image resampling.

Finally, we evaluate and compare our retargeting results to previous work and linear scaling in a user study with 121 subjects.

The video-retargeting framework presented in this thesis allows for high quality video reformatting of streaming content. It computes in near real-time a pixel-accurate warp considering many non-linear content aware and user controlled retargeting constraints. It also achieves high output quality by introducing a novel EWA based rendering technique.

1.2 Disparity Mapping for Stereoscopic 3D

Stereoscopic 3D is on the cusp of becoming a mass consumer product. Cinemas show an increasing number of movies produced in 3D, TV channels are beginning to launch 3D broadcasts of sports events, and companies are offering 3DTV sets and Blu-ray 3D players. But despite these technological advances, the production of natural and comfortable stereoscopic content is still a great challenge.

The fundamental problem lies in the complex interplay of human visual perception and the restrictions of display devices [HR02; Hof+08]. As a consequence, visual content must be adapted to the peculiarities of particular application scenarios. Diverse studies and psychophysical experiments have revealed fundamental limitations of current stereoscopic display devices [Hof+08]. While today's 3D display technology can recreate the effect of vergence (rotation of both eyes in opposite directions to maintain binocular vision), other important depth cues, such as accommodation (change of focus), cannot be faithfully reproduced as the resulting image is being displayed on a flat surface. This conflict has severe consequences; when displaying a close object on a distant screen, the strong negative disparity may result in an uncomfortable viewing experience and can cause temporary diplopia, the inability to fuse stereoscopic images. These effects are a major problem in practical 3D movie production. Content optimized for a standard 30 foot cinema screen will look completely different on a TV screen or a handheld display, and individual viewers can have vastly different viewing preferences. Auto-stereoscopic displays, i.e. displays that do not require viewers to wear 3D glasses conceptually require significantly more views than the two provided by a stereoscopic content pipeline. Current auto-stereoscopic display requires often up to as much as 30 views from a given scene. Hence, controlling and adapting disparity to the viewing situation as well as the ability of generating additional views for glasses-free displays is of central importance to the widespread adoption of stereoscopic 3D [SH09]. In addition, movie directors often employ (local) depth manipulation as an artistic and narrative device. All these issues have led to a complex set of best practice rules in the industry for how to film and display stereoscopic movies [Men09; Neu09]. Implementing these rules requires considerable expertise on how to control disparity during filming and post-production.

A further significant problem is the realization of these guidelines in practice. Once stereo footage is recorded, it is no longer possible to alter relevant parameters such as camera baseline or disparity range. In principle, techniques for depth-image-based view interpolation (DIBR) [Zit+04] could be

employed, but these methods tend to involve tasks such as estimating camera parameters, dense stereo reconstruction, and inpainting of occluded scene content. These are under-constrained and computationally complex problems, which cannot yet be solved with the necessary accuracy and robustness for general scenes and classical 2-view stereo footage. Therefore, movie and video producers have to resort to labor intensive and extremely costly manual editing of disparities (e.g., by compositing content from multiple stereo rigs of varying baseline). While this approach is expensive (but possible) in post-production for some scenarios, it is prohibitive for live broadcast where modifications of the disparity range have to be performed on the fly.

In this thesis we introduce and discuss *disparity mapping operators*. These operators are based on central aspects of disparity in stereoscopy. We review these aspects from a perceptual point of view and discuss the resulting implications and requirements for stereoscopic content production and display. Our operators then formalize these insights, and are the basis for a general framework for stereoscopic retargeting and disparity editing.

We then describe how image domain warping can be utilized to apply disparity mapping operators to stereoscopic footage. This is a conceptually much simpler than DIBR based methods, but a all the more practical and powerful new way for changing the depth impression of 3D content. With careful analysis we found that we only have to introduce a small set of additional constraints to or IDW framework to ensure consistent and content-adaptive remapping of the disparity range according to the chosen mapping operators. In contrast to previous works like traditional view interpolation, our method requires only a sparse set of stereo correspondences, which can be computed with sufficient robustness. We additionally improve the saliency measurement from 2D retargeting to be stereoscopic-aware. We will show how we naturally support manual disparity editing, which results in large artistic freedom and therefore integrates well into existing production workflows. With disparity aware image domain warping (3D-IDW) central problems of existing view interpolation methods such as camera calibration, accurate dense depth, and inpainting are avoided.

We demonstrate the versatility and practical relevance of our operators and warping technique on various types of stills and video. In particular, we present several applications of our method to central problems in stereo production: automatic disparity correction of live broadcast, nonlinear disparity editing and temporal disparity correction for movie post-production, retargeting of stereo footage to different display sizes, and 2D to 3D conversion of video.

As extension to 3D-IDW we show how additionally approximated views

from just two given input views can be generated. This has big practical relevance for autostereoscopic displays. It is not practical to shoot complex movies with much more than two cameras. Also a large body of two-view content is already available. For current and future auto-stereoscopic displays it is therefore mandatory to provide a possibility to allow the consumption of stereoscopic content. We show that 3D-IDW could be used to convert two-view stereo footage to multi-view content. Autostereoscopic displays often have an even smaller depth-budget than stereoscopic displays. Therefore, the depth compression properties of nonlinear disparity mapping operators used in 3D-IDW are even more essential. Transmitting a large number of views would require a very high bandwidth. We show that transmitting only a small number of views and some 3D-IDW based meta-information, and re-synthesizing missing views on the client side is a practical solution in many real life situations.

1.3 Data Regularization for High Definition Video

An important question for many image-processing applications is how well do these methods scale to *video sequences*. A general class of image-based graphics problems is based on solving an energy minimization combining data constraints with a regularization term enforcing smoothness. Some important problems of this type are optical flow, disparity estimation, and colorization.

A main challenge for video applications is the temporal continuity of results. It is a very important requirement for preventing visible artifacts. However, many new and promising methods in this class focus only on single-frame examples, and even the benchmark Middlebury optical flow dataset (an application specifically dealing with video) provides only eight frames of video for temporal examples, and ground truth only for a single frame [Bak+11]. In contrast, the average shot length of many modern movies and TV is in the range of four to six seconds. As such, for image-based methods to be useful in video applications, they must be able to generate temporally consistent results over sequences on the order of *hundreds* of frames.

The main difficulty to achieve temporal consistency is that the regularization term enforcing spatial smoothness creates dependencies between pixels, often resulting in a large non-convex optimization problem. While single-frame solutions are often tractable, applying these to multiple frames often increases the size of the problem rapidly due to the inclusion of the temporal dimension and thus making direct extensions of these methods to video volumes computationally infeasible.

The main objective of the method presented in this thesis is to create a memory and computationally efficient solution that enables *practical* temporal consistency for long sequences. To achieve this, we trade off accuracy for efficiency, and solve a simpler approximation of the global optimization. We use well known similarities between these optimization problems and nonlinear partial differential equations (PDEs), where anisotropic diffusion is often used to find solutions.

While our approach solves an approximation of the global optimization, by introducing a temporal smoothness assumption we can constrain ambiguities that would exist in a single frame and achieve high quality, temporally consistent results despite the simpler formulation. In addition, when user input must be provided (such as in the form of scribbles), our temporal continuity assumption can reduce the required manual effort by propagating this information both spatially and temporally. We apply our method to a number of problems such as optical flow, disparity estimation, depth up-sampling, colorization, and saliency computation. Unlike many recent accelerated processing methods, our approach is conceptually simple and fast even without exploiting GP-GPU parallelism.

1.4 Principal Contributions

This thesis makes the following main contributions

- **An image domain warping framework with anti-aliased rendering.** We introduce a novel general image domain warping framework. We show how an image warping function can be effectively discretized at pixel resolution. We discuss how such a discretization can be used to formulate various constraints and so allowing controlling an image deformation. Given warp constraints and warp discretization, we present a novel specialized iterative multi-scale solver to compute constrained image deformations. Given an input image and an image warp we then present an computationally efficient and high quality anti-aliased warp rending method.
- **An efficient art-directable approach for video retargeting suitable for streaming content.** We present a video retargeting pipeline that achieves very high performance and can directly be applied to streaming content. The pipeline performs the required steps including automatic feature deduction, artistic control image domain warp computation and alias free rendering. We additionally propose a

set of meta-data which is embeddable into a video stream and allows content-provider a controlled retargeting result on remote user devices.

- **A discussion of stereoscopic perception and an introduction to disparity mapping.** By examining the desired stereoscopic properties for good 3D content we derive and discuss a set of disparity mapping functions that can map given disparity ranges to more preferred ranges.
- **An image domain warping method for changing the depth perception of stereoscopic content.** We show that image domain warping can successfully and efficiently be used to modify the depth impression of stereoscopic video content in a novel way. The proposed image domain algorithm does not require dense depth map or camera calibration parameters. It is therefore better suited for standard 3D production systems than previous depth image based rendering (DIBR) methods.
- **Applying image domain warping for auto-stereoscopic content production.** Auto-stereoscopic display devices require a large number of views of a scene to create a seamless glasses-free 3D experience. We show that image domain warping can be used to generate the required number of views from a single stereoscopic input sequence. We demonstrate that believable in-between views, as well as extrapolate views can be generated by image domain warping.
- **An efficient algorithm for video regularization and sparse data up sampling.** We introduce a new and efficient algorithm to filter, interpolate and extrapolate information in a video. We show that our unified method can be used to compute optical flow as well as filter or up-sample many kinds of data. Our novel method is content-ware in particular edge sensitive as well as motion-aware. We introduce efficient solutions to a large number of applications based on our proposed method.

1.5 Thesis Outline

This thesis is organized as follows: Chapter 2 discusses related work in the fields of image and video retargeting, post processing of stereoscopic 3D video sequences, multi-view generation, temporal filter, optical flow and sparse data up-sampling. Chapter 3 will discuss the fundamentals of image warp computation and rendering. In particular it will focus on the

mathematical description of an image domain warp and will discuss possible methods to compute a warp given some warping constraints. It will also describe different ways for rendering a warped image and it will introduce the modified EWA framework for video splatting.

In Chapter 4 we will then discus the entire pipeline for streaming based artist driven video retargeting. It will describe the warp constraints used for video retargeting and describe all secondary algorithms required for video analysis. It will continue by presenting different applications and results for video retargeting.

Chapter 5 applies image domain warping to the problem of changing depth perception of stereoscopic video content. It will first introduce disparity mapping operators and define the basic new warping constraints for depth manipulation.

In Section 5.3 we will along other use cases discuss how image domain warping can be applied for 2D to 3D conversion and auto-stereoscopic content creation.

The Chapter 6 will discuss how we can improve the computation of various image maps like saliency maps or depth maps. This is an important subtask for video retargeting and disparity mapping. We will also discuss how the same algorithm can be used to retrieve optical flow of a video and how it can be used for many other kinds of sparse data up sampling or content aware video filtering.

Chapter 7 concludes the thesis and summarizes its main contributions and potential future work.

1.6 Publications

In the context of this thesis, the following publications have been accepted.

[Krä+09] Philipp Krähenbühl, Manuel Lang, Alexander Hornung, and Markus H. Gross. "A system for retargeting of streaming video". In: *ACM Transactions on Graphics, Siggraph Asia* 28.5 (2009). DOI: 10.1145/1618452.1618472.

[Lan+10] Manuel Lang, Alexander Hornung, Oliver Wang, Steven Poulakos, Aljoscha Smolic, and Markus Gross. "Nonlinear disparity mapping for stereoscopic 3D". In: *ACM Transactions on Graphics, Siggraph* 29.4 (2010).

[Lan+12] Manuel Lang, Oliver Wang, Tunç Ozan Aydin, Aljoscha Smolic, and Markus H. Gross. "Practical temporal consistency for image-based graphics applications". In: *ACM Transactions on Graphics, Siggraph* 31.4 (2012), p. 34.

The following co-authored papers have been published during the time of this thesis. Many of them are extensions and variations of the algorithms presented in this thesis.

[Smo+10] Aljoscha Smolic, Yongzhe Wang, Nikolce Stefanoski, Manuel Lang, Alexander Hornung, and Markus H. Gross. "Non-linear warping and warp coding for content-adaptive prediction in advanced video coding applications". In: *ICIP*. 2010, pp. 4225–4228.

[Far+11] Miquel Farre, Oliver Wang, Manuel Lang, Nikolce Stefanoski, Alexander Hornung, and Aljoscha Smolic. "Automatic content creation for multiview autostereoscopic displays using image domain warping". In: *ICME*. 2011, pp. 1–6.

[Smo+11a] Aljoscha Smolic, Peter Kauff, Sebastian Knorr, Alexander Hornung, Matthias Kunter, Marcus Müller, and Manuel Lang. "Three-Dimensional Video Postproduction and Processing". In: *Proceedings of the IEEE* 99.4 (2011), pp. 607–625.

[Smo+11b] A Smolic, S Poulakos, S Heinzle, P Greisen, M Lang, A Hornung, M Farre, N Stefanoski, O Wang, L Schnyder, et al. "Disparity-aware stereo 3d production tools". In: *Visual Media Production (CVMP), 2011 Conference for*. IEEE. 2011, pp. 165–173.

[Wan+11b] Oliver Wang, Manuel Lang, M. Frei, Alexander Hornung, Aljoscha Smolic, and Markus H. Gross. "StereoBrush: Interactive 2D to 3D Conversion UsingDiscontinuous Warps". In: *SBM*. Ed. by Tracy Hammond and Andrew Nealen. Eurographics Association, 2011, pp. 47–54. ISBN: 978-1-4503-0906-6.

[Wan+11c] Yongzhe Wang, Nikolce Stefanoski, Manuel Lang, Alexander Hornung, Aljoscha Smolic, and Markus H. Gross. "Extending SVC by Content-adaptive Spatial Scalability". In: *ICIP*. 2011, pp. 3493–3496.

[Gre+12a] Pierre Greisen, Manuel Lang, Simon Heinzle, and Aljoscha Smolic. "Algorithm and VLSI Architecture for Real-Time 1080p60 Video Retargeting". In: *Eurographics / ACM SIGGRAPH Symposium on High Performance Graphics* (June 2012), pp. 57–66.

[Sch+12] Lars Schnyder, Manuel Lang, Oliver Wang, and Aljoscha Smolic. "Depth image based compositing for stereo 3D". In: *3DTV-Conference*. 2012, pp. 1–4.

[SLS12] Nikolce Stefanoski, Manuel Lang, and Aljoscha Smolic. "Image quality vs rate optimized coding of warps for view synthesis in 3D video applications". In: *ICIP*. 2012, pp. 1289–1292.

[Ste+13] Nikolce Stefanoski, Oliver Wang, Manuel Lang, Pierre Greisen, Simon Heinzle, and Aljoscha Smolic. "Automatic View Synthesis by Image-Domain-Warping". In: *IEEE Transactions on Image Processing* 22.9 (2013), pp. 3329–3341.

CHAPTER 2

Related Work

2.1 Video Retargeting

The important problem of adapting images or video to different formats [Set+05; Kno+07] has been addressed in various ways in the literature. A variety of methods have been investigated to remove unimportant content by cropping or panning [Che+03; LG06]. The required visual importance of image regions can, for example, be estimated by general saliency measures [IKN98; GMZ08] or dedicated detectors [VJ04]. Limitations of these automatic techniques can to some extend be alleviated by manual training [DDN08]. Such adaptation, however, does not provide high level control with respect to the scene composition, which is a central feature of the algorithm presented in this thesis.

A different class of approaches removes unimportant content from the interior of the images or video [AS07; RSA08]. These techniques compute a manifold seam through the image data in order to remove insignificant pixels. While these approaches have shown very promising results for automatic retargeting they are still subject to significant conceptual limitations. Since the seam removes exactly one pixel per scanline along the resized axis large scale changes inevitably result in seams cutting through feature regions. In addition, the removal of pixels without proper reconstruction and bandlimitation results in visible discontinuities or aliasing artifacts.

The techniques that come closest to our own approach compute a non-uniform image warp to the target resolution without explicit content removal. The key idea of these methods is to scale visually important feature regions uniformly while permitting arbitrary deformations in unimportant regions of the image. This idea, for instance, has been utilized for feature-aware texturing [GSC06]. Here, a coarse deformation grid ensures that features rotate and scale only while non-feature regions follow a global, pre-defined warp. More sophisticated constraints on the warp, specifically designed for resizing images, have been proposed in the optimized scale-and-stretch approach [Wan+08]. The resulting warp preserves feature regions well for even significant changes of the aspect ratio. Similar concepts have been employed for image editing [SMW06] or 3D mesh resizing [Kra+08]. However, the coarse resolution of the deformation grid restricts the available degrees of freedom considerably, making it difficult to preserve small scale features. In contrast, our entire computational framework operates on the pixel level and thus utilizes the degrees of freedom to the maximum extend possible.

Content-driven *video* retargeting [WGC07] raises a number of additional issues such as temporal coherence of the warp function. Wolf et al. rescale an input video stream subject to constraints at the pixel resolution. Their technique is not capable of scaling important image content like, e.g., the optimized scale-and-stretch approach [Wan+08], since it tries to retain the original size of features. This strategy produces very plausible results for video containing human characters. At the same time, however, the approach produces excessive crops of the input so that the overall scene appearance is compromised. The performance of this method can be further improved by using shrinkability maps [ZHM08] which provide more directability, but are still limited with respect to the supported constraints.

To the best of our knowledge, none of the prior art considers high level, art directable control over the process, nor do they handle signal processing issues emerging from the resampling stage. Our work provides novel solutions to those important problems and represents the first approach to video retargeting that addresses the full problem domain.

2.2 Disparity Mapping

Stereoscopic 3D production and display for movies or 3DTV is a challenging multi-disciplinary field ([Red+02; Smo+11b; EU13]), combining basic research on binocular vision and perception, camera and display technologies, as well as cinematography and art.

The capabilities of our visual system and depth perception have been the topic of numerous works and experiments in research on human vision [BJ80; HR02]. One fundamental limitation is the range of disparities. As an example, we are unable to perceive extremely close and distant objects at the same time in 3D due to the large disparity range on our retina. Interestingly, however, our visual system still has quite strong abilities to compensate for inconsistent stereo cues, e.g., [Ste+00].

The rising popularity and recent developments of 3D display technology (e.g., [MP04]) requires a reinvestigation of perceptual limitations in the context of the technological capabilities. Most of the current 3D display technology is based on displaying a stereo image pair on a flat screen. This approach reproduces stereo cues such as vergence, but neglects other important depth-cues like accommodation. It has been shown that this discrepancy between accommodation and vergence yields problems such as distorted perception or visual fatigue [Hof+08; Lam+09], and considerable research efforts are invested to minimize these issues [SN00; Ake+04].

In stereoscopic content production, the most important tool to address such discrepancies between stereo cues is to adapt the range of disparities, i.e., the depth of a scene [Men09; SH09]. Besides pure adaption, however, control over scene depth is also an important artistic tool. Correspondingly there exists a complex set of cinematographic guidelines and rules on best practice in 3D movie making [Men09], as well as some prior work that allows for manually-driven disparity editing in specific application scenarios [PBP00; FSK03; WS08]. However, a rigorous formalization of these principles for disparity editing under consideration of perceptual as well as production-related issues has not been achieved yet. Inspired by our and others work on content retargeting and tone mapping [Rei+05; Wey+07; Wan+08] we present a solution for general *nonlinear disparity mapping operators* for stereoscopic 3D in Section 5.1.

Also on the technical level, disparity control of filmed stereoscopic video is a highly non-trivial problem, since novel views have to be generated that reflect the desired depth structure of the scene. The classical approach to this problem has been to perform image-based view interpolation (DIBR), which either requires a very large number of densely sampled input images or additional accurate depth maps to achieve high quality results [Gor+96; LH96; Sha+98; Zit+04; Cri+07; Kim+08; Smo+08; Ble+09]. One example of commercial software that uses image-based view interpolation for stereo editing is Ocula [Fou10]. These types of view interpolation involve a large number of computationally complex problems such as camera calibration [PKG99], accurate depth, inpainting and rendering. Due to this complexity,

fully automatic, sufficiently robust and accurate methods for cinematographic production and display adaptation are not available yet. There are also some techniques that provide a simplified manual interface for creating 3D scenes from video [Hen+07], or generating stereographic sequences from single view input [GWC09], but these methods require either calibration, static scenes, and manual tuning or dense depth estimation respectively. For small scale interpolation, image-based view morphing is an alternative [SD96; Mah+09]. The great advantage of these methods is that they directly work in image space without the complex reconstruction and rendering. However, they are not suitable for general adaption of the scene's global depth, since they do not support the required nonlocal consistency constraints.

Recently, methods based on warping have shown to be powerful tools for complex operations on images and video which preserve the realism of the original input, including camera stabilization [Liu+09], optimizing image content [CAA09], and video retargeting. Inspired by our own work in the field IDW we present a novel technique for *stereoscopic image warping* in Section 5.2 which enables complex disparity editing of existing stereoscopic 3D footage.

2.3 Efficient Temporal Video Regularization

In Chapter 6 we addresses a generalized method used to solve a wide array of problems, and a full review of all applications is outside the scope of this section. Instead, we present an overview explaining how our work relates to selected relevant approaches.

2.3.1 Global Optimization

Traditionally, image-based data + regularization problems are solved by defining a combined error term and computing a global minimum. Optical flow is commonly found by iteratively solving a linear system of equations [HS81a]. Disparity estimation (stereo) methods often use a similar error formulation, which is solved with graph cuts, simulated annealing, or other such approaches [SS02], and a closed form solution was presented for colorization [LLW04]. These methods all require optimizing a large number of free variables, which leads to large memory requirements and computationally expensive convergence. Both of these problems are made worse when dealing with large (HD) images and video sequences. We present a much

simpler approximation that allows us to process long, high resolution video shots.

2.3.2 Optical Flow

Implicit computation of optical flow is an integral part of applications that produce temporally consistent results, as it is used to model the motion of objects between frames. Many high-performing modern methods still use some variation of the original Horn Schunck formulation [HS81a], incorporating modified data terms and iterative, pyramid-based solutions [ZBW11]. Bilateral filtering has been integrated into an iterative variational framework, replacing the traditional anisotropic diffusion step [Xia+06]. These methods operate only on neighboring pairs of video frames, are computationally expensive, and cannot easily enforce temporal continuity, a main focus of our work.

Other recent efficient methods filter a matching-cost-volume and select a minimum-error underlying surface to compute discretized optical flow and disparity estimates [Rhe+11]. Like our approach, these methods also use edge-aware filtering, although for different purposes. One of their main advantages is that the optical flow is computed locally, allowing for GPGPU parallelization. However, constructing cost volumes for full video sequences is impractical given the size of the discrete label space. Our approach allows us to work in-place on the video with lower memory requirements and does not require discretization of the solution space, which can lead to artifacts.

2.3.3 Temporal Consistency

Temporal stability has been recognized as a significant open problem. Proposed solutions include sliding windows [Hos+11; Vol+11] and Kalman filtering [HOK11]. With these methods, each output frame is still computed locally, and greedy decisions can lead to temporal inconsistencies. In addition, selection of the window size is an important parameter balancing computational requirements with temporal smoothness. Our approximation avoids this trade-off as we can process entire video shots.

For colorization, methods have directly solved global optimization problems on video volumes using pre-computed optical flow to model frame-to-frame relationships [LLW04; Bha+10]. These approaches can generate very high quality output, but are computationally expensive and do not scale well to high resolution images or long video sequences.

2.3.4 Filtering

Common approaches for edge aware filtering use bilateral filter weights [TM98], or a local linearity assumption [HST10]. Efficient methods for computing these use advanced space representations and GPU parallelism [CPD07; ABD10]. However, these optimizations are not as well suited for filtering with large kernel sizes, which is a necessity of our method.

Another class of edge-aware filtering uses weights based on the geodesic-distance [Cri+10; GO11], which performs pixel mixing inversely proportional to the distance over the $\mathbb{R}^5$ ($RGBXY$) image manifold, as opposed to the ℓ_2 norm (used for bilateral filtering). We use geodesic-distance filtering, as it more closely approximates the diffusion-model in an anisotropic heat equation, and as shown later in Fig. 6.7 yields better results for our application. Specifically, we develop an extension to the domain transform which reduces 2D image filtering to a series of 1D operations.

One important application for bilateral filtering is sparse data up sampling. This is often investigated for sensor fusion applications ([G+08; DT05; Dol+10]). We will show results of our uniform method also for such applications.

CHAPTER 3

Image Domain Warping

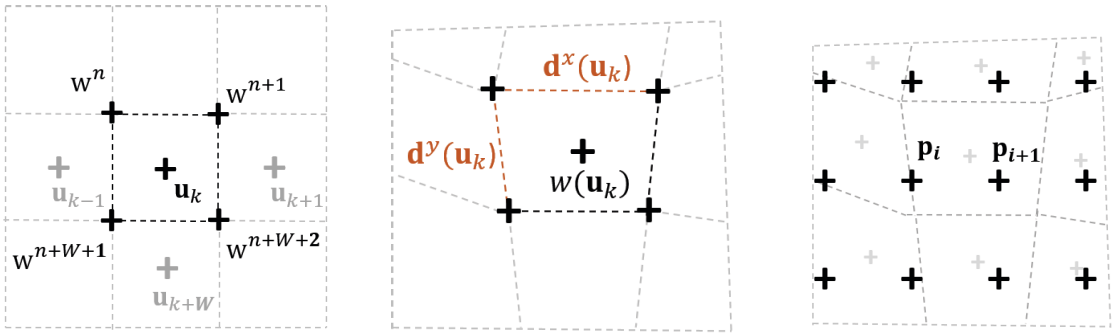
In this chapter, we will introduce the concept of image domain warping (IDW). This will give us the basic concepts and theory required for the different applications in the remainder of this thesis. In particular, we first introduce grid and pixel based image warping. We will show how image domain warping can be formulated as an energy minimization with a given set of target constraints. By introducing some common constraints for many applications we layout the general technique for introducing additional constrains in later chapters.

In the second section, we will discuss rendering of a warped grid. We introduce the modified elliptic weighted average (EWA) framework for 2D rendering. It is used to map a given input image according to a given per-pixel warp to an output domain. One important property of EWA rendering is that it reduces the amount of aliasing introduced by resampling operations. In addition, we will compare EWA rendering to other rendering methods such as bilinear texture mapping.

3.1 Mathematical Foundation of Constraint Image Warping

We define an image warp as a function deforming space

$$\begin{aligned} \mathbf{w} &: \mathbb{R}^2 \to \mathbb{R}^2 \\ (\mathbf{u}) &\mapsto w(\mathbf{u}) \qquad\qquad \text{for } \mathbf{u} := (x,y)^T \in \mathbb{R}^2 \end{aligned} \tag{3.1}$$



(a) unity grid (b) warped grid (c) re-sampling

Figure 3.1: *Discretization of a warp. Initially the pixel distance and thus the distances between all $\mathbf{u}_k$ is one. In the undeformed grid (a) the grid vertices w^n have also distance one to their neighbors. In the warped grid (b) the grid cell edge lengths are an approximation to the partial derivative in x and y dimension ($d^x(\mathbf{u}_k)$, $d^y(\mathbf{u}_k)$). The warped pixel position is at the center of the deformed grid cell. For rendering the warped input image is re-sampled at uniform positions $\mathbf{p}_i$ (c)*

We call locations in the original space u. Warp w maps every such location to a new location $\mathbf{p} = w(\mathbf{u})$ in the deformed space. We write $w_x(\mathbf{u})$ for the x-coordinate of the target location $\mathbf{p}$. Moreover, for video manipulation it is often essential to have dependent consecutive warping functions for sequential frames. Therefore, to express such an requirements we will in some parts add the time dimension t and therefore get a warp definition for a frame at time t as $(x,y)_t \mapsto w(x,y,t)$.

The objective for the various applications is to find such a warp function w that fulfills specific constraints to achieve a desired effect (i.e. change aspect ratio, or modify stereoscopic disparity). Additionally, we practically always require the warp function to be smooth or in some applications at least piecewise smooth.

The warp function is computed by an energy minimization that enforces the application specific constraints. We will discuss this in Section 3.3

3.2 Warp Discretization

The continuous warp definition Eq. (3.1) over $\mathbb{R}^2$ is of limited practical use. Firstly, the input domain (i.e. an digital image) is already discretized and secondly there is no simple closed form continuous solution for the required

minimization of the warp. Therefore, it makes sense to discretize the warp formulation early on. This enables us to also specify the warp constraints in this discrete setting.

We first reduce the degrees of freedom of the continues warp by assuming local smoothness and approximating it with its first order tailor expansion at discrete positions $\mathbf{u}_k = (x_k, y_k)^T$

$$\mathbf{p} = w(\mathbf{u}) \approx w(\mathbf{u}_k) + \mathbf{J}_k(\mathbf{u} - \mathbf{u}_k) \tag{3.2}$$

$\mathbf{J}_k$ is the Jacobi matrix i.e the first partial derivatives at position $\mathbf{u}_k$. Therefore for a discrete position $\mathbf{u}_k$ in the input domain the warp can be represented by 2x2 matrix and a 2D vector. It is a natural choice to use the initial pixel positions of the input image for the positions $\mathbf{u}_k$. This results in 6 degrees of freedom per pixel position $\mathbf{u}_k$. Another interpretation for Eq. (3.2) is an affine deformation of a unity square around each pixel position $\mathbf{u}_k$. Thereby, the Jacobian encodes the scaling, shearing and rotation of the unity quad. The full deformation is then described by the quad corner vertices.

Another important and reasonable assumption is that neighboring pixel stay neighbors after deformation and that the distortion is similar. Therefore, we can share the edges of neighbor quads. Given an image with $W \times H$ pixels we can fully describe any possible image warp by $(W+1) \times (H+1)$ quad corner positions. We call those grid corner positions vertices of the grid $\mathbf{w}^n$ where n is in $[0 \dots (W+1)(H+1)]$. Please note that we use n to name grid vertices and k to enumerate pixels. Every pixel $\mathbf{u}_k$ is surrounded by four grid vertices. Neighbor pixels always share two grid vertices. Fig. 3.1 shows how such a warp grid is constructed.

Given the discrete warp in a grid structure, its derivatives can be approximated with finite differences. The distance between two pixels in the undeformed image is one. Therefore, initial edge length of the grid cell is also one. The finite differences are directly represented by the grid edge length around a pixel. This is shown in Fig. 3.1. The approximation of the derivatives at a pixel location k are:

$$\begin{aligned}
\frac{\partial w_x(\mathbf{u}_k)}{\partial x} \approx d_x^x(\mathbf{u}_k) \approx w(\mathbf{u}_k + (1,0)^T) - w(\mathbf{u}_k) \approx \\
\approx \mathbf{w}_x^{n+1} - \mathbf{w}_x^n \qquad (3.3) \\
d_y^x(\mathbf{u}_k) \approx \mathbf{w}_y^{n+1} - \mathbf{w}_y^n \qquad (3.4) \\
d_x^y(\mathbf{u}_k) \approx \mathbf{w}_x^{n+W+1} - \mathbf{w}_x^n \qquad (3.5) \\
d_y^x(\mathbf{u}_k) \approx \mathbf{w}_y^{n+W+1} - \mathbf{w}_y^n \qquad (3.6)
\end{aligned}$$

$d_y^x(\mathbf{u}_k)$ is thereby the y-coordinate of the finite difference approximation of the derivative towards x at the location u_k. The warping grid width is $(W+1)$ and the grid vertex neighbor to the right is therefore $n+1$ and the neighbor below is $n+W+1$. See Fig. 3.1(a,b) for an illustration. We call $\mathbf{d^x}(\mathbf{u}_k) = (d_x^x(\mathbf{u}_k), d_y^x(\mathbf{u}_k))^T$ the gradient towards x at location $\mathbf{u}_k$.

3.3 Constraints and Energy Minimization

The previous section we introduced the warp function and its discretization. In the following, we show how such a warp can be constrained to enforce the desired attributes of a given application.

Knowing the discretization of the warp, we can formalize the warp deformation constraints. These constraints are expressed as energy terms. The weighted squared sum of all energy terms for all constraints is then minimized over the degrees of freedom of the warp, i.e. the quad corner positions. For image processing applications, the energy terms are often separated int a set of data terms and a set of smoothing terms. Following that approach we can write the minimization as

$$\min_w E(w) = E_{data}(w) + E_{smooth}(w) \tag{3.7}$$

$E_{data}(w)$ enforces application specific constraints, which usually are sparse across the whole image. Terms in $E_{smooth}(w)$ smooth the data over the entire optimization domain, which in our case is the deformation grid. $E_{smooth}(w)$ is thereby not only important to get a smooth result, but is essential to get a defined system where each degree of freedom is represented in the minimization. In the following, we will first describe how the warp constraints can be constructed. Then, we will show how the energy term can be efficiently minimized using a general multi-scale solver that optimizes both terms simultaneously.

3.3.1 Warp Constraints

The terms $E_{data}(w)$ and $E_{smooth}(w)$ are the sum of individual quadratic energy functionals. The data terms are used to necessitate specific properties at given positions in the warp. The smoothness constrains define the neighborhood behavior and try to propagate sparse effects to the rest of the image. Data terms are more application specific whereas the smoothing terms are comparable for many applications

Common Smoothness Constraints

In general, it is important that the warp deformations are not arbitrary. Most image processing applications try to avoid visible distortions. Therefore, the Jacobian should be close to a scaling matrix, often even close to the identity matrix. We enforce the Jacobean to be a scaling matrix at all input pixels u_k.

$$\mathbf{J}_k = \begin{bmatrix} d_x^x(\mathbf{u}_k) & d_x^y(\mathbf{u}_k) \\ d_y^x(\mathbf{u}_k) & d_y^y(\mathbf{u}_k) \end{bmatrix} \sim \begin{bmatrix} s_x^k & 0 \\ 0 & s_y^k \end{bmatrix} \tag{3.8}$$

For every position $\mathbf{u}_k$ we can therefore derive four smoothness energy terms, by using grid edge length as forward finite differences to discretize the Jacobian.

$$\forall k :$$

$$c_{s1}^k := \alpha_{s1} \cdot \left(d_x^x(\mathbf{u}_k) - s_x^k \right)^2 \tag{3.9}$$

$$c_{s2}^k := \alpha_{s2} \cdot \left(d_x^y(\mathbf{u}_k) - 0 \right)^2 \tag{3.10}$$

$$c_{s3}^k := \alpha_{s3} \cdot \left(d_y^x(\mathbf{u}_k) - 0 \right)^2 \tag{3.11}$$

$$c_{s4}^k := \alpha_{s4} \cdot \left(d_y^y(\mathbf{u}_k) - s_y^k \right)^2 \tag{3.12}$$

Those constraints enforce the desired Jacobian in a least square fashion. $\alpha_{s1} \dots \alpha_{s4}$ are the weights indicating the importance of the constraints during optimization. Depending on the application, these weights can be user-defined values or derived from the image itself. We discuss these weights in the context of the respective applications in the following chapters. The function to optimize for the entire image is the sum of all constraints at all locations.

$$E_{smooth}(w) = \sum_k c_{s1}^k + c_{s2}^k + c_{s3}^k + c_{s4}^k \tag{3.13}$$

Data Constraints

Data constraints are much more application specific than smoothness terms. The most simple constraints enforce precomputed final positions of specific image regions. Therefore, they directly impose a (small) number of warp values at positions $\mathbf{p}_n$:

$$c_d^n := \alpha_d \left| \mathbf{w}^n - \mathbf{p}_n \right|^2 \qquad E_{data}(w) = \sum_n c_d^n \tag{3.14}$$

The weight α_d again defines the strength or importance of a data constraint. Please also not, that because this constraint enforces properties in the target spaces we used the corresponding notation (output location $\mathbf{p}$ and enumeration n). Most application have more complex data constraints. Some will be in similar form but with a more complexly derived α_d or $\mathbf{p}$. Typical example is the disparity constraints of the disparity mapping application described in Chapter 5 Section 5.2.3.

Other applications constraints may be more similar to the a smoothness constraints but more sparsely enforced. The constraint enforcing uniform scaling in salient regions Section 4.2.1 is such an example. However, in some of the applications the constraints are even non-linear in w. Section 4.2.2)

3.3.2 Energy Minimization

Given all constraints and corresponding energy terms we can now setup the energy minimization. The best strategy for the energy minimization greatly depends on constraints required by the applications. With only linear constraints one can use any standard linear least square optimization such as direct solvers. However, for many applications the constraints only allow for an non-linear based solver. In such cases we use a simple gradient decent iterative solver.

In the following, we will briefly discuss both classes of solvers.

Linear Least Square

If all constraints are linear, we can rearrange all energy terms to the following matrix form by putting all free variables of the function w into the vector $\mathbf{w}$:

$$E_l(\mathbf{w}) = \sum_k (c_l^k)^2 = \|\mathbf{A}_l\mathbf{w} - \mathbf{b}_l\|^2 \tag{3.15}$$

The warp w that minimizes the energy terms is found at $\frac{\mathrm{d}}{\mathrm{d}\mathbf{w}} E_l(\mathbf{w}) = 0$. Therefore, with

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{w}} E_l(\mathbf{w}) = 2\mathbf{A}_l^T(\mathbf{A}_l\mathbf{w} - \mathbf{b}_l) = 0, \tag{3.16}$$

we derive the well known normal equations

$$\mathbf{A}_l^T\mathbf{A}_l^T\mathbf{w} = \mathbf{A}_l^T\mathbf{b}_l. \tag{3.17}$$

Often the normal equation are constructed directly instead of using $\mathbf{A}_l$ and $\mathbf{b}_l$ to compute the products and transpositions. One can show that by using

$\mathbf{A}_l^T \mathbf{A}_l^T w = d^2/dw^2 E_l(w)$ and $\mathbf{A}_l^T \mathbf{b}_l = -d/dw E_l(w)$ one can directly fill in the elements of the normal equation [LH74]. The resulting system is highly sparse. Considering only the smoothness and data terms of Section 3.3.1 the system is diagonally dominant with only five non-zero elements per row (the current position and it is four direct grid neighbors). Note that some of the applications we are going to discuss in this thesis will introduce more non-zero elements, however in all applications the system remains sparse and diagonally dominant. This is particularly true for the very sparse data constraints in E_{data}.

Non-Linear Energy Terms

So far we have only discussed constraints that lead to linear energy terms. However, in most applications we will also require non-linear energy terms. To support those constraints we have to rely on sparse iterative non-linear optimization schemes. Here, we discuss quickly how to derive the parameters for such an optimization.

We differentiate two kinds of non-linear constraints. Hard constraints and non-linear terms in w.

Hard constraints Firstly, most applications have hard constraints. Hard constraints for example specify an absolute position for some grid corners. For video retargeting where the goal is to resize an image to a new size, the grid corners along the image edge are fixed to the target position. Such hard constraints can be introduced by substitution. Thereby, the constraints unknowns are removed from the optimization problem and replaced by constant itself.

Secondly, most applications require a simple inequality constraint that avoids fold-overs of the grid. To prevent fold overs one can require that $\mathbf{w}_x^n < \mathbf{w}_x^{n+1}$ and $\mathbf{w}_y^n < \mathbf{w}_y^{n+(W+1)}$ i.e. the x and y coordinates in the grid have to be strictly monotonically increasing. To enforce those constraints we rely on our iterative solver Section 3.4.2.

Non-linear constraints Some applicatons, most notable the image retargeting application described in Chapter 4, introduce real non-linear constraints. This means that the resulting energy term of the constraint is non-linear within the grid position unknowns. Typical example is the line constraint in Section 4.2.2. It allows to enforce that multiple grid vertices are only warped

in such a way that they together on a virtual line. Having such energy terms requires us to use a non-linear iterative solver.

3.4 Solver Algorithms

Depending on the application and its constraints different solver algorithms can be utilized. In the following we will introduce the different algorithms used throughout this thesis. The individual application sections will then specify, and refine important details of the used solver.

3.4.1 Direct Solvers

For applications that employ linear constraints with simple hard constraints only, a direct linear least squares solver can be used. Direct solvers use algorithms like Gaussian elimination, or techniques like Cholesky-, QR- or simply LU-decomposition to directly obtain an exact solution for the normal equations. Such direct solvers have a complexity in the order of $O(n^3)$ for general dense matrices, but even sparse systems do not significantly reduce complexity. For diagonally dominant sparse band matrices like in our setup the complexity is in the order of $O(nW^2)$ [Gre13]. Thereby, n is the number of unknowns in the system, in our case its in the order of twice the number of pixels of the involved images. W is the image width. For most of our image warping applications this is prohibitively expensive both in regards of computation time and memory consumption. Therefore, even with linear constraints we often need to use iterative solvers.

However, in some applications or during experimentation and verification it is useful to use a direct solver. In such cases we may reduce the warp grid resolution to be much coarser than the pixel accuracy. Computing the warp deformation only for every tenth or twentieth pixels makes the problem much more feasible. The per- pixel positions can then be found with bilinear interpolation within a warping grid cell. See the rendering by texturing in Section 3.5.2 for more details.

Implementation for Direct Solvers

We relied on well optimized third-party solver libraries for direct solvers. In most application we directly used the Matlab backslash operator. Often however other parts of the application had already an optimized C++ implementation. In such cases we used Matlab native code interface to invoke the

backslash operator directly from within our C++ implementations. We found that Matlab has the most advanced algorithms for deciding which specific direct solver implementation to use.

3.4.2 Iterative Solver

Iterative solvers work by approximating the result of a given problem by stepwise reducing the approximation error over multiple iterations. In our energy-minimizing problem, we update the current approximated minimum during each iteration by going downhill along the energy manifold. More specifically, we update the current minimum by a vector proportional to the negative of the gradient of the energy function at the respective approximated minimum. This scheme is known as gradient descent minimization. Besides its much lower complexity this scheme can be easily parallelized for our 2D grid based problem. Additionally, non-linear constraints can be introduces easily as long as we can compute their gradients. However, gradient descent methods are not generally converging to a global minimum and may converge slowly especially when the application requires high accuracy results. Nevertheless, we found that for image domain warping and its typical constraints we could achieve good approximations in a reasonable number of iterations (fast enough for interactive results).

Given our the structure of the problem as 2D grid and given that all postulated constraints use the finite difference approximation we find that all update steps can be given in the following form:

$$\mathbf{w}^{new} = \sum_{\mathbf{w} \in \mathcal{N}(\mathbf{w}^{old})} q_{\mathbf{w}}(\mathbf{w}^{new}) \cdot \mathbf{w} \tag{3.18}$$

$\mathbf{w}^{new}$ is the new value for one grid vertex. It is computed by a weighted average of the (old) values of all its neighbors including its own old value. The set of old neighbor grid vertices is called $\mathcal{N}(\mathbf{w}^{old})$. For the individual warping application one has to derive the neighbor weighting factors $q_{\mathbf{w}}(\mathbf{w}^{new})$ at all grid locations. The general approach for all non-linear energy terms is to analytical compute the first derivative at each grid vertex. From the finite difference approximations and bz setting the derivative to zero, we can then derive the necessary weights and neighbor sets. In Section 4.3.1 we show how the weights and the set of neighbors can be derived for two examples. Please note that the weights $q_{\mathbf{w}}(\mathbf{w}^{new})$ normally depend on the image content and are therefore different for each grid vertex.

During these iterative updates we can also enforce hard inequality constraints introduced in Section 3.3.2. Generally, inequalities would yield to much more

complicated optimizations (e.g. linear programming), however we found that we can achieve good results by simply modifying our iterative solver. After every iteration we fix potential grid fold overs in the warp by forcing back vertices to position where the inequality is fulfilled.

Multi-Scale and Parallel Implementation

Inspired by the Gauss-Seidel iteration scheme [YJ88] and multi-grid [Hac86] methods we implemented a parallel, multi-scale gradient descent method on the GPU.

Multi-Scale We tried to reduce the problem of getting stuck in local minima as well as improving the general performance of gradient descent by introducing a multi-scale iteration scheme. This is inspired by the well known multi-grid solvers [Hac86] but slightly simplified. Our 2D grid and our constraints allows to directly solve in the problem domain at all scales instead of solving on the residuals as standard multi-grid would. To distinguish we call this variation multi-scale.

With the muli-scale approach we take advantage of the 2D grid setup of our problem. Instead of evaluating each iteration for all unknowns in the system, we solve the problem at different scales. First we compute a hierarchy of multiple coarser grids based on the input per-pixel grid. At each coarser scale we only have a fourth of the unknowns compared to the next finer scale. We start by computing the lowest approximation of the gradient descent minimum at the coarsest grid. Then we scale this coarse result bilinearly up to the next finer grid resolution and use it as initial guess to start the another gradient descent iteration. When we reach the finest per-pixel level we stop the iteration after we have achieved sufficient accuracy. A downside of this approach is the difficulty of deriving the update steps for the lower resolution grids. One has to adapt $q_{\mathbf{w}}(\mathbf{w}^{new})$ and the neighbors of the set $\mathcal{N}(\mathbf{w}^{old})$ of equation Eq. (3.18). This can be done similarly to the step multi-grid methods call restriction [Hac86].

Parallel It is essential to parallelize an algorithm to take full advantage of current computing hardware. We found that our gradient descent update steps can be easily parallelized by utilizing the 2D grid structure of the problem. We simply compute the update step Eq. (3.18) for each grid vertex independently and in parallel to every other grid vertex. We found that in practice we do not need to serialize the neighbor access i.e., making sure that

the update rules only access the old value of its neighbors. In our experiments, we achieved good convergence even if the neighbor pixel values were already updated. Additionally, most of the memory accesses needed are very local (i.e the grid neighbors and update step weights) which allows using efficient cache and local memory architectures efficiently.

We therefore can easily spawn as many threads as we have unknowns i.e., grid vertices in the system. Each thread can independently update its assigned vertex iteratively until the system converges.

3.5 Rendering of a Warped Image

After the image warp that follows the application constraints most faithfully has been obtained, the corresponding output image has to be rendered. To achieve this, a given input image has to be transformed as specified by the warp function to obtain the output image. For most application the image warp given for rendering is a non-linear and spatially varying function over the image domain. As both input and output images are represented as discretely sampled 2D signals, our rendering has to be analyzed as a non-linear spatially varying resampling process. Fig. 3.2 (c) shows a comparison of trivial rendering and our optimized version which we are going to introduce in section Section 3.5.3.

The non-linear warp function alters the spectral energy distribution of the image and therefore the following necessary re-sampling step potentially maps too high-frequency energy to an output with much lower frequency spectrum. Such spurious frequencies have to be eliminated from the output signal by proper bandlimitation for aliasing-free images. Therefore, in this section we first mathematically describe the rendering process. We then present two rendering techniques. First, we discuss the most commonly use texturing inspired grid rendering approach and then we propose a novel forward mapping approach based on EWA theory.

3.5.1 Rendering as Resampling Process

Given an 2D input image I_I with discrete intensities specified at uniformly distributed distinct positions $\mathbf{u}_k \in \mathbb{N}^2$ and given an image warp function $\mathbf{p}_k = w(\mathbf{u}_k)$ that maps every pixel position to a new arbitrary location $\mathbf{p}_k$, during rendering we have to find the output intensities at new uniformly distributed discrete positions $\mathbf{p}_i$ as defined by the regular pixel grid of the output image I_O.

First, we reconstruct the continuous input function from discrete pixel sampling

$$f_s(\mathbf{u}) = \sum_{k=0\ldots W_i H_i} I_I^k g_I(\mathbf{u} - \mathbf{u}_k). \tag{3.19}$$

This results follows from using the discrete input samples I_I^k and convolving it with the a 2D interpolation function g_I. Given the continuous input intensity function f_s one can derive the continuous output function in the output domain by utilizing the inverse backwards warp function w^{-1}

$$f_c(\mathbf{p}) = \sum_{\forall k} I_I^k g_I(w^{-1}(\mathbf{p}) - \mathbf{u}_k). \tag{3.20}$$

The final discretized output image I_o can now be derive by sampling f_c at all necessary discrete locations $\mathbf{p}_i$. To avoid aliasing one has to bandlimit the function f_c with an anti-aliasing filter $h(\mathbf{p})$ before discretizing. Therefore, with convolution we get the following solution for the output intensities at all pixels i:

$$\begin{aligned} I_O^i &= f_{c,aa}(\mathbf{p})|_{\mathbf{p}=\mathbf{p}_i} = (f_c \otimes h)(\mathbf{p})|_{\mathbf{p}=\mathbf{p}_i} \\ &= \int_{\mathbb{R}^2} \sum_{\forall k} I_I^k g_I(w^{-1}(\boldsymbol{\omega}) - \mathbf{u}_k) h(\mathbf{p} - \boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \Big|_{\mathbf{p}=\mathbf{p}_i} \end{aligned} \tag{3.21}$$

Eq. (3.21) is a closed form solution for rendering an output image given an input image, an arbitrary warp function w, an interpolation-filter g_I, and an anti-aliasing filter h. In practice this equation is only of limited practical use. The inverse mapping w^{-1} is complicated or even impossible to derive analytically, and because of the non-linear coordinates transformation and the infinite support of the optimal anti-aliasing and interpolation filter the convolution can not be computed efficiently. Therefore, we now discuss two approximations to the optimal rendering equation Eq. (3.21).

3.5.2 Backwards Mapping by Utilizing Graphics Hardware

In the backwards mapping approach one iterates over all output pixels locations $\mathbf{p}_i$ and tries to estimate the output color by looking "backward" to the input image. Theoretically, this requires the inverse image warp function w^{-1}. This function however is not at all trivial to derive for general, non-linear image transformation. Nonetheless, one can approximate it by using a forward warped grid and interpolate the actually required backward lookup coordinates with linear interpolation. This approach is the standard method used in 3D rendering for mesh texturing and therefore easily implemented on a GPU.

Grid-based Texturing

As explained in Section 3.2 we computed the application dependent image warp function with a discretized grid approximation. This discretized grid can now directly be used to implement the rendering on the GPU. We use the grid vertex positions as standard vertex input to a typical real-time rendering pipeline (i.e OpenGL) and split every quad of the grid in two triangles. Additionally, for each grid vertex we set a texture coordinate. It is simply the coordinate of vertex in the undeformed grid.

The original input image is used as the rendering texture. The graphics pipeline automatically interpolates a lookup texture coordinate during rasterisation for each output pixel. This bilinear interpolated coordinate is used to "texture fetch" the color in the original image. Thereby, standard texture filtering is applied to suppress aliasing.

GPU's typically use mip-mapping and in the best case anisotropic filtering to avoid aliasing. However, with the highly non-linear and spatially varying warp those approximations to the optimal filer h are not sufficient. The discrete mip-mapping levels can introduce visible artifacts neighboring regions that are deformed very differently. Although better suited, anisotropic filtering can also not be controlled sufficiently enough on a GPU, i.e. skewed and rotated grid deformations cannot be considered.

However, texture based grid rendering is a simple to implement approach that often yields sufficiently good quality. Also,if the warp grid is not pixel accurate backwards mapping is better suited than forward mapping.

3.5.3 Forward Mapping with EWA Splatting.

We present a novel forward mapping based rendering approach for 2D image warping. The work of [GH86] and [Zwi+02] introduces splatting of elliptical weighted average filter kernels (EWA) for 3D rendering. Splatting is a forward mapping method and it works by computing for every discrete input sample the potential contribution to every output sample. The influence region in the output domain effected by one input samples is called a splat. [GH86] introduces splats with an elliptically, arbitrarily oriented influence falloff function for 3D rendering. [Zwi+02] further shows that if interpolation filter and anti-aliasing filter are assumed to be Gaussian low pass filters one a can derive an approximated single Gaussian splat for transformations typical in 3D rendering (affine and perspective projection) While originally being

devised for 3D rendering, we tailor this method to the case of 2D image synthesis for high quality, aliasing-free output.

Rendering Function of a Grid Based Warp

The general rendering function equation Eq. (3.21) only requires the Image warp function w to be invertible. We discussed in Section 3.2 that the warps are computed with a discretized and linearized grid approximation. We can utilize this fact and rearrange equation Eq. (3.2) to

$$\mathbf{u} = w^{-1}(\mathbf{p}) \approx \mathbf{J}_k^{-1} \cdot (\mathbf{p} - w(\mathbf{u}_k)) + \mathbf{u}_k \tag{3.22}$$

By substitution this into the warp rendering equation Eq. (3.21) and by applying the integral sum rule to replace the order of summation and integration we get (in the neighborhood $w(\mathbf{u}_k)$):

$$\tilde{f}_{c,aa}(\mathbf{p}) = \sum_{\forall k} \int_{\mathbb{R}^2} I_I^k \cdot g_I(\mathbf{J}_k^{-1}\boldsymbol{\omega}) \cdot h(\mathbf{p} - w(\mathbf{u}_k) - \boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \tag{3.23}$$

It is important to see that we can now easily derive the influence of every input location k for every output location p by just considering all the summands individually. Therefore we have found the forward mapping splatting function for every input pixel.

EWA Filter Design

The mathematically optimal filter for both the interpolation (g_I) and anti-aliasing filter h is a perfectly sharp "brick-wall" low-pass filter. However, such a filter is impractical because it exhibits infinite support in the space domain (and therefore infinitely large splats). Simply limiting the support domain introduces however serious artifacts and is therefore not a good approximation to the optimal filter. The EWA framework uses two-dimensional Gaussian filters as better approximation to the ideal low-pass filter, especially because Gaussians exhibit some additional desirable properties. Applying an affine transformation to a Gaussian filter or convolving two Gaussians yields again a Gaussian filter. Given these two properties we can similarly to [Zwi+02] derive a simple closed form splatting function.

We build on [Zwi+02], who showed that both the interpolation filter as well as the anti-aliasing filter can be combined into a single Gaussian splat for 3D

point rendering. We derive the same result for 2D rendering by replacing the two filters in Eq. (3.23) with Gaussians of the form

$$G_{\mathbf{V}}(\mathbf{d}) := \frac{1}{2\pi|\mathbf{V}|^{1/2}} \cdot e^{-0.5\mathbf{d}^T\mathbf{V}^{-1}\mathbf{d}} \tag{3.24}$$

we get the following Gaussian filters:

$$h(\mathbf{d}) := G_{\mathbf{V}_a}(\mathbf{d}), \tag{3.25}$$

$$g_I(\mathbf{J}_k^{-1}\mathbf{d}) := G_{\mathbf{V}_i}(\mathbf{J}_k^{-1} \cdot \mathbf{d}) = \frac{1}{\mathbf{J}_k^{-1}} G_{\mathbf{J}_k\mathbf{V}_i\mathbf{J}_k^T} \tag{3.26}$$

$\mathbf{V}_i = diag(\sigma_{i,x}^2, \sigma_{i,y}^2)$ and $\mathbf{V}_a = diag(\sigma_{a,x}^2, \sigma_{a,y}^2)$ are the covariance matrices for the reconstruction filter and the anti-aliasing filter as described in the EWA framework. It defines the size of the splats. We choose the variance of the interpolation filter in such a way that effects on neighbor pixels are minimal. The variance of the anti-aliasing filter is defined by the output sampling frequency. Building upon our work, [Gre13] further investigates the properties of the EWA filter design and optimizes those parameters to optimally follow a the optimal sinc function. Additionally, his work introduces location depended anti-aliasing filter, whereas we used a constant filter.

By combining both filters we can derive a single Gaussian splatting function $f_{splat,k}$ for one input sample k

$$\mathbf{V}_k = \mathbf{J}_k\mathbf{V}_i\mathbf{J}_k^T + \mathbf{V}_a \tag{3.27}$$

$$f_{splat,k}(\mathbf{p}) := I_I^k * \frac{1}{\mathbf{J}_k^{-1}} \cdot G_{\mathbf{V}_k}(\mathbf{p} - w(\mathbf{u}_k)) \tag{3.28}$$

Implementation of EWA Splatting

We have implemented the complete EWA rendering algorithm on the GPU for fast real-time output. The first step is to compute the transformed interpolation kernels co-variance matrix $\mathbf{V}_k$ from the warp grid. Fig. 3.2 shows the undeformed and deformed grid and corresponding splat. As described in Section 3.2 we can derive $\mathbf{J}_k$ by simple finite difference for each grid cell. This can be done in parallel on the GPU with a CUDA kernel. Additionally, we compute the pixel center $w(\mathbf{u}_k)$ as the a average of the four surrounding grid vertices (Fig. 3.1). The result, namely the covariance matrix $\mathbf{V}_k$ and the warped pixel center position $\mathbf{p}_k = w(\mathbf{u}_k)$ as well as the pixels un-warpped position $\mathbf{u}_k$ are then directly stored to a OpenGL vertex buffer.

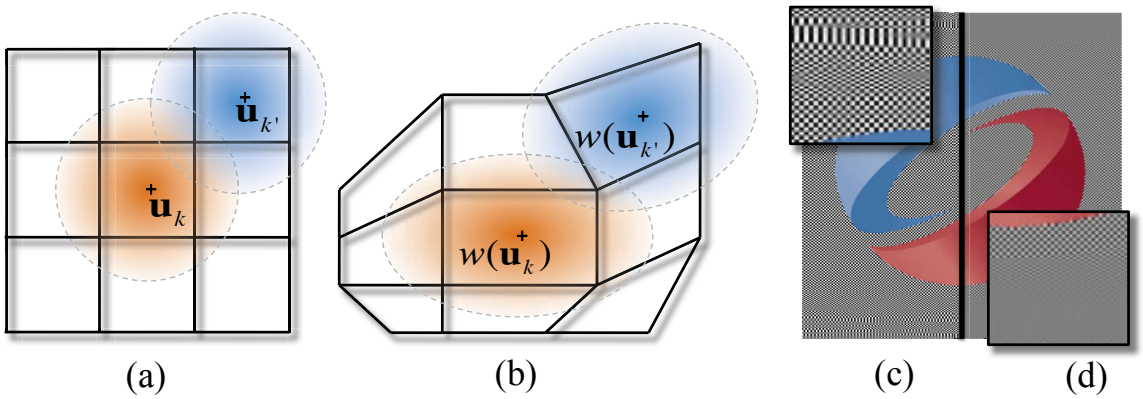


Figure 3.2: *Illustration of the warp discretization and rendering. (a) The undeformed pixel grid and basis functions. (b) After computation of the warp. (c) Rendering of a warped image without anti-aliasing. (d) Result of our algorithm for EWA video rendering.*

The second step is performs the actual splatting. The information in the vertex buffer ($\mathbf{u}_k$, $\mathbf{p}_k$, $\mathbf{V}_k$) together with input image I_i as texture is piped into a programmable OpenGL graphics pipeline. In the vertex buffer a splat bounding box is computed. It is that axis-aligned box in which the GPU rasterizer and fragment shader compute the splat influence for each output pixel. Due to the rapid decay of the influence of the Gaussian, and for performance reasons we use a bounding box with a $boxsize = (d * max(\sigma^2_{k,x}, \sigma^2_{k,y}))^{1/2}$ with $d = 10px$. Additionally, the undeformed location $\mathbf{u}_k$ is used to sample the color from input texture and the result is passed to the fragment shader. The fragment shader actually computes the contribution by evaluating Eq. (3.28). The graphics blending stage is used to add the influences of all splats together. Because, the approximations in Eq. (3.23) and the cutoff of the Gaussian introduce some errors and additional normalization step is required. In a second rendering pass the summed up color for every output pixel i is divided by sum of all influence weights of all splats that overlap that pixel.

3.6 Summary

In this chapter we introduced image domain warping. We laid out the foundation to compute image warps given some constraints that describe a desired deformation. We formalized the image warp as a discrete grid with the grid vertices as unknowns of a large energy minimization. The energy minimization can be solved efficiently on a GPU with an iterative multi-scale solver. We introduced a fast and novel anti-aliased forward-rendering technique by adapting EWA 3D point splatting to 2D image warping.

In the following sections of this thesis we will now apply this warping framework to various applications.

CHAPTER 4

Video Retargeting

Figure 4.1: *Two examples displaying results from our interactive framework for video retargeting. The still images from the animated short "Big Buck Bunny" compare the original with the retargeted one. The pictures on the right show two different rescales. Thanks to our interactive constraint editing, we can preserve the shape and position of important scene objects even under extreme rescalings*

In this chapter we show how to apply image and video domain warping for the application of video retargeting. Video retargeting is the process in which a given video is reformatted and adapted to a new output display aspect ratio. We present an integrated and interactive framework which combines key frame based constraint editing with numerous automatic algorithms for video analysis. This combination gives content producers high level control of the retargeting process. The central component of our framework is a non-uniform, pixel-accurate warp to the target resolution which considers

automatic as well as interactively defined features. Automatic features comprise video saliency, edge preservation at the pixel resolution, and scene cut detection to enforce bilateral temporal coherence. Additional high level constraints can be added by the producer to guarantee a consistent scene composition across arbitrary output formats. For high quality video and interactive frame rates we utilize the novel 2D EWA rendering framwork introduced in Section 3.5.3. Our method seamlessly integrates into postproduction and computes the reformatting in realtime. This allows us to retarget annotated video streams at a high quality to arbitary aspect ratios while retaining the intended cinematographic scene composition. For evaluation we conducted a user study which revealed a strong viewer preference for our method.

4.1 Overview

The aim of our method is to resize a video stream, i.e., a sequence of images $I_0, I_1, \ldots, I_t : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ in a context-sensitive and temporally coherent manner to a new target resolution. This means that we have to find a spatio-temporal warp $w_t : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, i.e., a mapping from coordinates in I_t to new coordinates in O_t. Fully automatic warps most often fail to retain the actual visual importance or output style intended by a producer or director. Therefore, our approach combines automatic detection of features and constraints with a selection of simple but effective tools for interactive key frame annotation to compute the warp function.

The conceptual components of the resulting retargeting pipeline are illustrated in Fig. 4.2. Given a current frame I_t of the video stream the system automatically estimates visually important features based on image gradients, saliency, motion, or scene changes. Next, a feature preserving warp w_t to the target resolution is computed by minimizing an objective function $E(w)$ which comprises different energy terms derived from a set of feature constraints (see also Chapter 3). The first set of specialized application defined energy terms for video retargeting measure local quality criteria such as the uniformity of scaling of feature regions, the bending or blurring of relevant edges, or the spatio-temporal smoothness of the warp (Section 4.2.1). In addition we include the producer's interactively annotated high level features and constraints with respect to the global scene composition. This input refers to the position, shape or saliency of an image region. These constraints integrate seamlessly into the overall optimization procedure (Section 4.2.2).

The warp w_t is computed in a combined iterative optimization including all target terms of the energy function (see Section 4.2.3). All computational

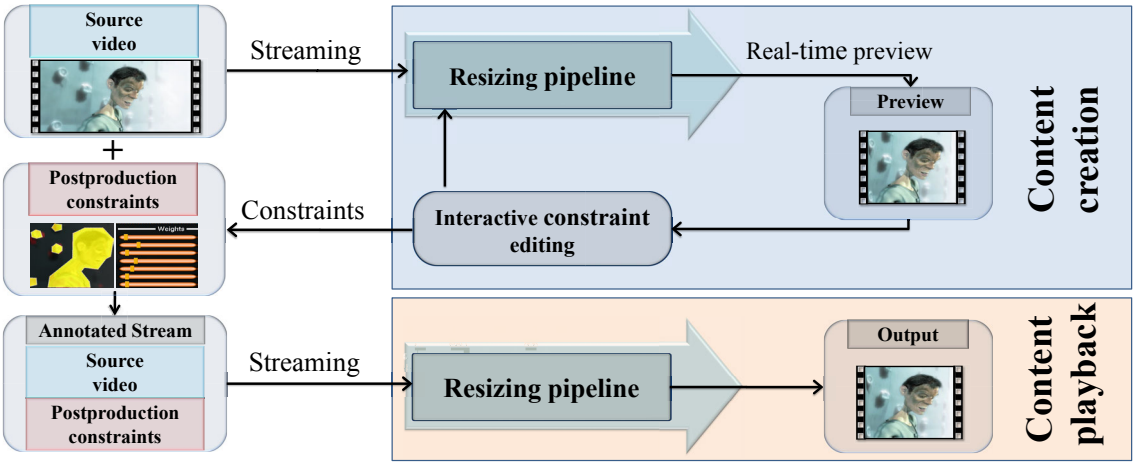


Figure 4.2: *Postproduction pipeline for key frame editing. Output is a sparsely annotated video stream suitable for real-time retargeting.*

steps are performed at pixel resolution in order to faithfully preserve even small scale image features. The rescaled output frame O_t is then rendered using hardware accelerated per-pixel EWA splatting. This technique ensures real-time performance and minimizes aliasing artifacts (Section 3.5.3).

Since our method works in real-time and thus provides instant visual feedback, video editing and resizing can be accomplished in a fully interactive content production workflow (see Fig. 4.2). After editing, the high level constraints can be stored as sparse, time-stamped key frame annotations and streamed to the end-user along with the original input video. This compound video stream supports a viewing experience that matches the one intended by the video producer as closely as possible. In the following sections we will first describe all the application dependent constraints of the retargeting method and then discuss relevant specialized implementation details in Section 4.3.

4.2 Image Warp for Video Retargeting

An ideal warp w_t for the retargeting application must resize input video frames I_t according to user-defined scale factors s_w and s_h for the target width and the height of the output video, respectively. In addition, it must minimize visually disturbing spatial or temporal distortions in the resulting output frames O_t and retain the interactively defined constraints from the content producer. We therefore extend our image domain warping pipeline to be temporally aware as well as introduce specific automatic and interac-

tive constraints. This section introduces those additional constraints and quickly summarizes the warp computation. For detail discussion of our image warping pipeline please refer to Chapter 3.

4.2.1 Automatic Features and Constraints

Previous work offers different approaches to distinguish important regions from visually less significant ones. Most of this work focuses on low-level features from single images. We draw upon some of these results and employ a combination of techniques for automatic feature detection. In addition, we propose a number of novel warp constraints at different spatio-temporal scales that improve the automatic preservation of these features considerably.

Saliency Map and Scale Constraints

A common approach to estimate the visual significance of image regions is the computation of saliency maps. Literature provides two main strategies for generating such maps. The first class of methods estimates regions of general interest bottom-up and is often inspired by visual attentional processes [IKN98]. These methods are generally based on low level features known to be important in human perception like contrast, orientation, color, intensity, and motion. A second class of top-down methods uses higher level information to detect interesting regions for particular tasks. Examples include detectors for faces or people [VJ04].

Since our method focuses on real-time retargeting of *general* video, we designed a GPU implementation of a bottom-up strategy [GMZ08]. This method utilizes a fast, 2D Fourier transformation of quaternions [ES07] to analyze low-level features on different scales. The resulting real-time algorithm to compute the saliency map $F_s : \mathbb{R}^2 \rightarrow [0,1]$ captures the spatial visual significance of scene elements.

Another important visual cue is motion. Therefore, processing video requires additional estimates of the significance based on temporal features. For example, a moving object with an appearance similar to the background is classified as unimportant by spatial saliency estimators for single images. When considering the temporal context, however, such objects are stimulating motion cues and thus are salient. We take temporal saliency into account by computing a simple estimate of the optical flow [HS81b] between two consecutive video frames. The resulting motion estimates are added to the

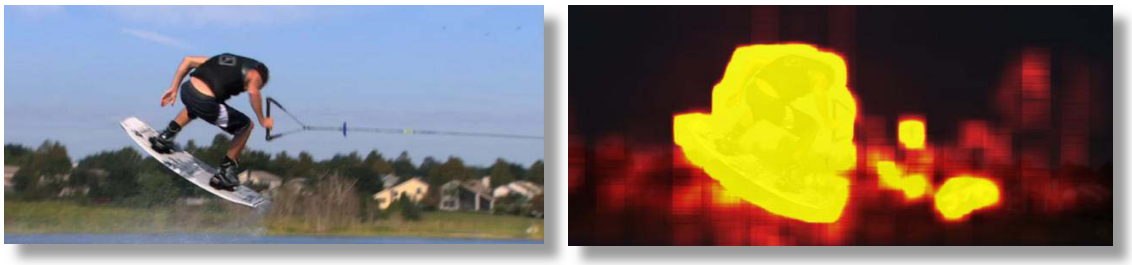


Figure 4.3: *Spatio-temporal saliency map F_s.*

global saliency map F_s and provide additional cues for the visual importance of scene elements. Fig. 4.3 displays an example.

In order to preserve salient image regions represented by F_s during the resizing process we define the constraints below for the warp function: To simplify the notation we will remove index t from now on for non-temporal constraints. On a global level w must satisfy a target scale constraint in order to meet the intended scaling factors s_w and s_h. Let w_x denote the x-component of the warp w. The global scale constraint yields

$$\frac{\partial w_x}{\partial x} = s_w \quad \text{and} \quad \frac{\partial w_y}{\partial y} = s_h. \tag{4.1}$$

In feature regions of F_s, however, a uniform scaling factor s_f must be enforced to preserve the original aspect ratio:

$$\frac{\partial w}{\partial x} = \begin{pmatrix} s_f \\ 0 \end{pmatrix} \quad \text{and} \quad \frac{\partial w}{\partial y} = \begin{pmatrix} 0 \\ s_f \end{pmatrix}. \tag{4.2}$$

In previous methods the scale factor for feature regions across an image may change arbitrarily. We enforce a *single* scale factor s_f, which ensures that all features are subject to the same change of scale. This retains global spatial relations and the overall scene composition much more faithfully.

As described in Section 3.2 we discretized the warp at the pixel level and rewrite the above constraints as a least squares energy minimization with finite differences. The constraints are of the form of a typical smoothnesses constrains E_smooth as introduced in Section 3.3.1. However, they are slightly modified depending on the location u_k. For salient regions we enforce that the Jacobian matrix is a uniform scaling matrix with just one global scaling factor s_f for all locations. For all other regions we only enforce that the diagonal of the Jacobian should represent a scaling that brings the image close to the new aspect ratio; we do not restrict the off-diagonal elements.

With $d^x(\mathbf{u})$ and $d_x^x(\mathbf{u})$ as the finite difference approximations of $\frac{\partial w}{\partial x}$ and $\frac{\partial w_x}{\partial x}$ at a pixel $\mathbf{u}$, the global scale energy term according to Eq. (4.1) is

$$E_g = \sum_{\forall k} (d_x^x(\mathbf{u}_k) - s_w)^2 + \left(d_y^y(\mathbf{u}_k) - s_h\right)^2, \tag{4.3}$$

and the uniform scale constraint Eq. (4.2) for salient regions becomes

$$E_u = \sum_{\forall k} F_s(\mathbf{u}_k) \left(\left(d^x(\mathbf{u}_k) - \begin{pmatrix} s_f & 0 \end{pmatrix}^T\right)^2 + \left(d^y(\mathbf{u}_k) - \begin{pmatrix} 0 & s_f \end{pmatrix}^T\right)^2 \right). \tag{4.4}$$

Additionally, we made s_f a free parameter as well. This means, one does not specify beforehand how much the important regions are scaled. We only enforce that all salient regions are uniformly scaled with one single value. This, however makes the constraint in Eq. (4.3) non-linear. Fortunately, it is just one single additional degree of freedom and a closed form of the partial derivative $\partial E(w)/\partial s_f$ can be found. Therefore, the update step necessary to optimize s_f after each iteration can be computed.] See Section 4.3 for details.

Edge Preservation One of the most simple indicators for small scale image features are edge detectors based, e.g., on image gradients. An edge detector itself does not constitute a sophisticated indicator for general visual importance. Its combination with our pixel level warp, however, allows us to design local constraints for feature edge preservation. In our current implementation an edge map F_e is computed using a standard Sobel operator [GW02] (see Fig. 4.4). More sophisticated edge detectors could of course be integrated easily.

Bending of prevalent feature edges F_e can be avoided by a spatial smoothness constraint following [WGC07]:

$$\frac{\partial w_x}{\partial y} = \frac{\partial w_y}{\partial x} = 0. \tag{4.5}$$

We provide an additional constraint to avoid edge blurring or vanishing of detail, e.g., when enlarging an image (see Fig. 4.5). This can be achieved by enforcing similar image gradients for feature edges $\nabla I_t = \nabla(O_t \circ w_t)$ in order to preserve the original pixel resolution before and after the warp:

$$\frac{\partial w_x}{\partial x} = \frac{\partial w_y}{\partial y} = 1. \tag{4.6}$$

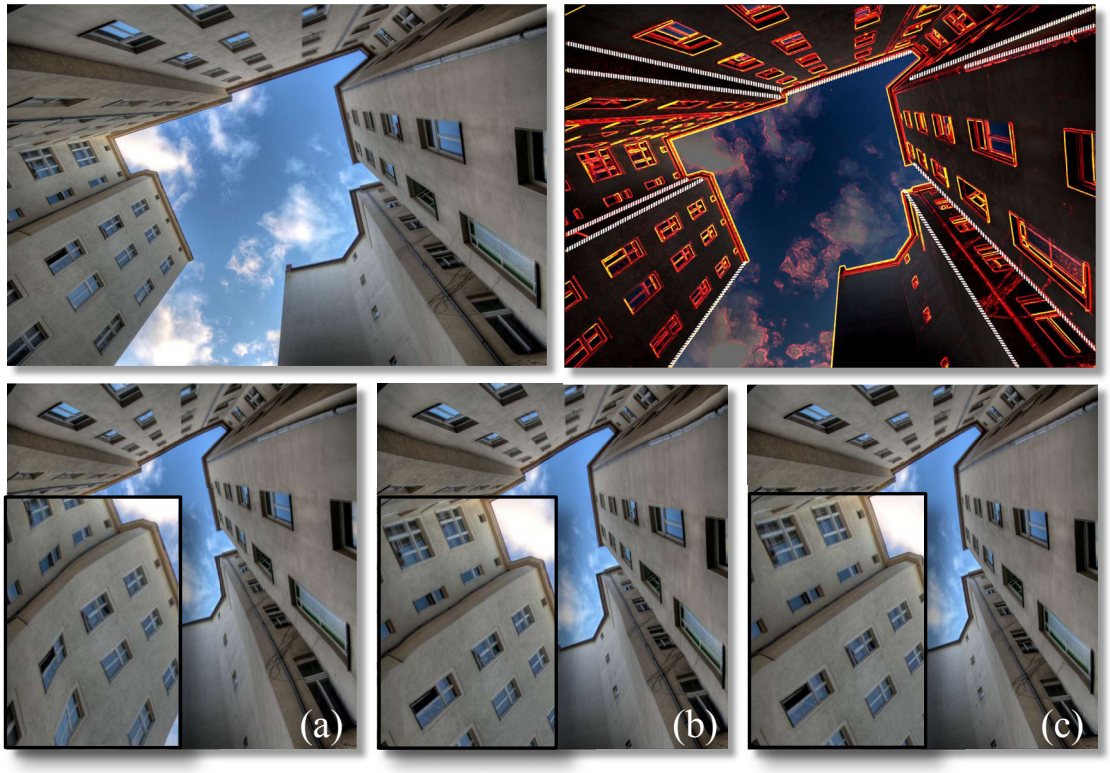


Figure 4.4: *Edge bending. The top row shows the original frame (left) and the edge map F_e (right) with additional, manually added line constraints (white). We compare the rescaling result of Wang et al. [Wan+08] (a) displaying considerable deformation of straight edges with a result (b) using our automatic constraints only. A further improvement can be achieved by manual annotation of line constraints (c).*

The corresponding bending energy and our novel edge sharpness energy for the warp optimization are similar to Eq. (4.3):

$$E_b = \sum_{\forall k} F_e(\mathbf{u}_k) \left(d_y^x(\mathbf{u}_k)^2 + d_x^y(\mathbf{u}_k)^2 \right) \quad \text{and} \tag{4.7}$$

$$E_s = \sum_{\forall k} F_e(\mathbf{u}_k) \left((d_x^x(\mathbf{u}_k) - 1)^2 + (d_y^y(\mathbf{u}_k) - 1)^2 \right). \tag{4.8}$$

Eq. (4.5) prevents bending of horizontal and vertical edges. However, in combination with Eq. (4.6) bending of diagonals is prevented as well. Note also that an image warp at pixel resolution is necessary in order to realize the sharpness constraint Eq. (4.6) effectively.

Bilateral Temporal Coherence Temporal coherence is an important albeit non-trivial issue in video retargeting. On the one hand, temporal stabilization

Figure 4.5: *Enlarged SIGGRAPH logo without (left) and with (right) our constraint for edge sharpness Eq. (4.6). Note the improved edge preservation and reduction of aliasing in the closeup on the right.*

is imperative in order to avoid jittering artifacts. On the other hand, the local and unilateral constraint

$$\frac{\partial w}{\partial t} = 0 \tag{4.9}$$

employed in previous work [WGC07] disregards the global nature of this problem: simply enforcing per-pixel smoothness along the temporal dimension does not take object or camera motion, nor discontinuities like scene cuts into account. An in-depth treatment of temporal coherence requires a pre-analysis of the full video cube and an identification of opposing motion cues. Since we are aiming at real-time processing with finite buffer sizes, we opted for the following approach which balances computational simplicity and suitability for streaming video.

First, an automatic scene cut detector based on the change ratio of consecutive edge maps F_e [ZMM95] detects discontinuities in the video. The resulting binary cut indicator Ft_c yields a value of 0 for the first frame of a new sequence and 1 otherwise. Using this indicator and Eq. (4.9) a bilateral temporal coherence energy for the warp computation (similar to the concept of bilateral signal filters) can be defined as

$$E_c = t_c \sum_{\mathbf{p}} d_t(\mathbf{u}_k)^2. \tag{4.10}$$

To account for future events (like characters or objects entering a scene) we perform a temporal filtering of the per-frame saliency maps F_s over a short time window of $[t, t+k]$ of the video stream. The filter thus includes information about future salient regions into the current warp and achieves a more coherent overall appearance. In practice, a small lookahead of $k = 5$ frames turned out to be sufficient in all our experiments. The introduced latency can be neglected. By utilizing our indicator t_c for scene cuts the saliency integration becomes aware of discontinuities in the video as well. In combination these two bilateral constraints effectively address local as well as global temporal coherence. This bilateral saliency integration is different

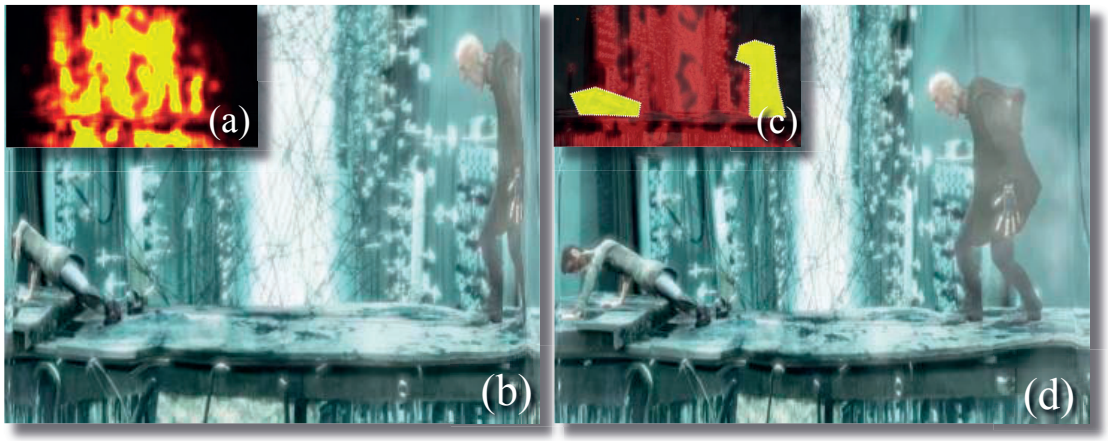


Figure 4.6: *(a) Automatic saliency estimators often cannot distinguish characters from detailed background. (b) As a result, the characters in the warped frame exhibit unnatural deformations. (c) With a simple interface the user can create polygonal importance masks in a few key frames and reduce the saliency of the background. (d) Utilizing this annotation and interpolation of the masks between key frames, the warp is able to retain the proportions of the characters much more faithfully during rescaling.*

from the previously introduced motion estimates, and it improves temporal processing significantly.

Besides the presented automatic constraints it is easily possible to add existing higher level feature estimators such as face detectors or others. However, the above combination of automatic detectors works very well on a broad spectrum of different video content without introducing too many parameters.

4.2.2 Interactive Features and Constraints

Although automatic features and constraints are required for a practical retargeting system, they share a number of limitations: first, automatic methods fail for insufficiently discriminating texture. This limitation can be addressed by simple editing of the corresponding feature maps. Second, automatic constraints are inherently limited in the representation of global shape constraints or, even more importantly, higher level concepts of scene composition. A simple example is illustrated in Fig. 4.4 where the warp bends building edges due to the locality of the edge bending constraint.

Manual editing and annotation of such user defined constraints is prohibitively cumbersome if done on a per-frame basis. Therefore, we borrow the

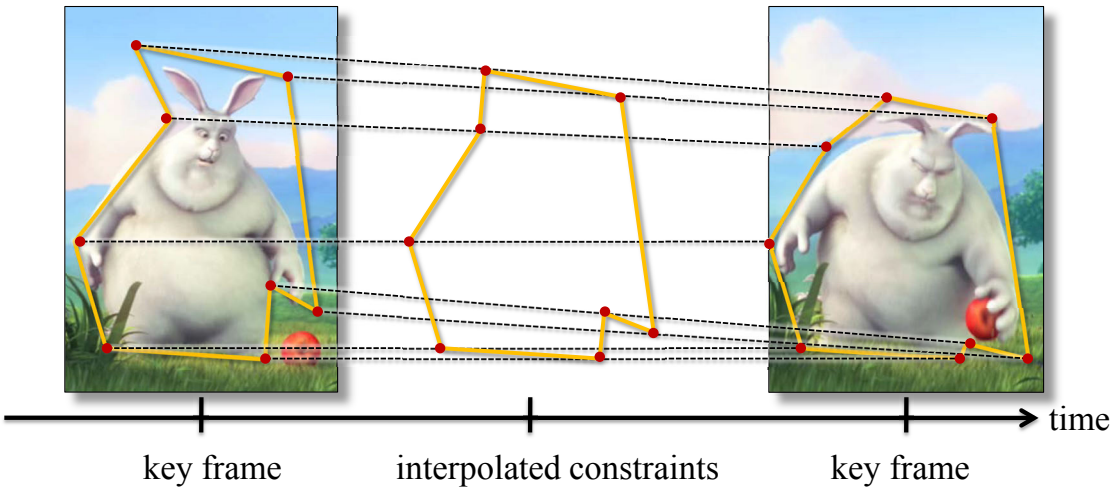


Figure 4.7: *Illustration of key frame based editing and interpolation of a polygonal importance mask. Our high level constraint editing and propagation is based on the same concept.*

well-established concept of key frame video editing and design a workflow that allows users to annotate constraints on a sparse set of key frames. As we will explain subsequently, these constraints will be propagated throughout the video. Fig. 4.7 illustrates the process. The depicted character has been marked as important by the user in two consecutive key frames. The shape of this annotated polygonal region is being interpolated linearly between the two key frames. Based on this concept we introduce the following set of simple and intuitive tools for manual warp editing.

Feature Maps and Key Frame Definition A simple, but powerful approach to guide the warp is the direct editing of the feature maps introduced in Section 4.2.1. Our system provides a simple drawing interface where the user can interactively select an arbitrary frame from the video, label it as a key frame and modify, e.g., the saliency map F_s by manually specifying the importance of individual image regions. Fig. 4.6 shows an example of this operation.

Object Position In particular for more complex scenes the realization of an intended visual composition often requires the specification of positional constraints for certain scene elements. Hard constraints [Wan+08], however, can introduce undesirable discontinuities when computing the image warp at pixel level as we do in our setting. Moreover, such hard constraints would

Figure 4.8: *Rescaled frames without (a),(c) and with (b),(d) a positional constraint for the rock. This interactively defined constraint allows us to preserve the relative position of scene elements within a frame, independent from the target aspect ratio.*

only be valid for a particular target size and aspect ratio and not allow for dynamic resizing of the video stream.

Instead we first let the user mark a region of interest R and then create a relative location constraint $\mathbf{loc} \in [0,1]^2$ for its center of gravity $\mathbf{cog}$ and with respect to the input image. During the optimization we recompute the center of gravity in each iteration i

$$\mathbf{cog}^i = n \sum_{\forall k} w^i(\mathbf{u}_k) \tag{4.11}$$

where n is a normalization factor and w^i corresponds to the warp computed in the i-th iteration. Next we optimize the following energy for each region R

$$E_P = (\mathbf{loc} - \mathbf{cog}_r^i)^2 \tag{4.12}$$

by adding the update vector $(\mathbf{loc} - \mathbf{cog}_r^i)$ to all pixels in R. Here, $\mathbf{cog}_r^i$ simply corresponds to $\mathbf{cog}^i$ converted to relative coordinates from $[0,1]^2$. Fig. 4.8 shows an example in which the user sets a positional constraint for a scene element.

Line Preservation Our visual perception is particularly sensitive to straight lines, such as edges of man-made structures. Automatic edge bending constraints as in Eq. (4.5) prevent bending locally, but cannot account for these structures on a global scope (see also comparison in Fig. 4.4). Hence, as a second high level constraint we provide means to preserve straight lines

globally. A line constraint is created by simply drawing a line represented as $l : sin(\alpha)x + cos(\alpha)y + b = 0$ in a frame of the input video. The system estimates the intersection of this line with the underlying pixel grid of the image, it assigns a corresponding coverage value $c(\mathbf{u}_k) \in [0, \sqrt{2}]$ and enforces

$$\sin(\alpha)w_x(\mathbf{u}_k) + \cos(\alpha)w_y(\mathbf{u}_k) + b = 0 \tag{4.13}$$

for each pixel $\mathbf{u}_k$ with $c(\mathbf{u}_k) > 0$. The objective function for the least squares optimization is

$$E_L = \sum_{\forall k} c(\mathbf{u}_k) \left(\sin(\alpha)w_x(\mathbf{u}_k) + \cos(\alpha)w_y(\mathbf{u}_k) + b\right)^2. \tag{4.14}$$

Updates of line orientation and position can again be computed from the derivatives of Eq. (4.14) with respect to α and b, similar to the estimation of s_f mentioned in Section 4.2.1. The effect of this constraint is displayed in Fig. 4.4.

It is important to note that the above constraints are defined in such a fashion that they remain valid for different aspect ratios of a retargeted video. Our real-time implementation enables users to instantly verify the results of the warp editing process for different target scales. Hence, the video producer can analyze whether the intended scene composition is preserved for the desired viewing formats.

4.2.3 Energy Optimization

The combined warp energy generated from all available target terms finally yields

$$E(w) = \underbrace{E_g + \lambda_u E_u + \lambda_b E_b + \lambda_s E_s + \lambda_c E_c}_{\text{Automatic constraints}} + \underbrace{\lambda_P E_P + \lambda_L E_L}_{\text{Interactive constraints}} \tag{4.15}$$

The minimization of this energy constitutes a non-linear least squares problem which is solved using an iterative multi-grid solver on the GPU (see Section 4.3 and Section 3.4.2). Note that our actual implementation allows for multiple interactive constraints. For boundary pixels of a video frame the respective coordinates are set as hard constraints.

Of the four weighting parameters λ controlling the automatic constraints, λ_u for uniform scaling of features was constantly set to $\lambda_u = 100$ for all our experiments. For the remaining three parameters we used default values $\lambda_b = 100$, $\lambda_s = 10$, and $\lambda_c = 10$ for most experiments. We will discuss the benefit of changing these parameters for different input like real-world scenes,

cartoons, or text in Section 4.4. For increased flexibility the influence of interactive constraints can be weighted on a continuous scale. However, we simply used a value of 100 for both parameters λ_P and λ_L in all corresponding examples.

4.3 Implementation

In order to achieve real-time performance we implemented our retargeting pipeline fully on the GPU, using CUDA [Buc07] for the feature estimation and energy minimization and OpenGL [SA06] for the EWA image synthesis. The different types of feature estimation techniques described in Section 4.2.1 can be transferred to the GPU in a straightforward manner. From a technical point of view the key components of our method are a multi-scale solver for computing the warp w_t and the EWA based rendering. The following two sections will discuss implementation details which we consider relevant for a reimplementation of the system.

4.3.1 Retargeting Multi-Scale Solver

The non-linear least squares minimization of E_w is solved with a multi-scale solver as introduced in Section 3.4.2. We will here shortly summarize the method and then describe with two examples the general way to find the necessary iterative update steps.

The retargeting warp optimization is implemented on the GPU. Only that way we can achieve the required performance. Additionally, all the necessary image analysis is already carried out using CUDA. Therefore, we can reduce slow GPU-CPU transfers by also implementing the warp optimization (and later rendering) on the GPU directly.

Gradient Descent Update Steps

Our iterative solver is a gradient descent solver Section 3.4.2. Therefore, during each iteration we update every unknown of the optimization with a value derived from the variables partial derivative.

We first look at one of the video retargeting smoothness terms e.g., E_g and the update steps for the warp grid corner x coordinates w_x. To find the update

step we compute the first partial derivative in w_x of E_g

$$\begin{aligned}\frac{\partial}{\partial w_x} E_g &= \frac{\partial}{\partial w_x(\mathbf{u}_k)} \sum_{\forall u'_k} \left(d_x^x(\mathbf{u}'_k) - s_w\right)^2 + \left(d_y^y(\mathbf{u}'_k) - s_h\right)^2 \\ &= 4 \cdot w_x(\mathbf{u}_k) - 2 \cdot w_x(\mathbf{u}_{k+1}) - 2 \cdot w_x(\mathbf{u}_{k-1}),\end{aligned} \tag{4.16}$$

and set it to zero to find the minimum

$$w_x^{new}(\mathbf{u}_k) = \frac{w_x^{old}(\mathbf{u}_{k+1}) - w_x^{old}(\mathbf{u}_{k-1})}{2}. \tag{4.17}$$

For the derivative we use the discretization of d_x as introduced in Section 3.2. We found that during each iteration we can compute the x position of every grid vertex by looking at the direct horizontal grid neighbors. Eq. (4.17) is already in the required form for our iterative solver Eq. (3.18).

It is important to note that we do have boundary constraints. In the video retargeting application the x coordinates of the most left pixels is set to 0 and the very right image edge pixel coordinates are set to new target width. This information propagates throughout the entire image. The above update equation only accounts for the E_g energy term. A similar derivation has to be done for all other energy terms for all free variables.

Other iteration updates for free variables can be found similarly. To have one more example, the update step of the optimal scaling factor for salient regions s_f can be derived as

$$\begin{aligned}&\frac{\partial}{\partial s_f} \sum_{\forall k} F_s(\mathbf{u}_k) \left(\left(d^x(\mathbf{u}_k) - \begin{pmatrix} s_f & 0 \end{pmatrix}^T\right)^2 + \left(d^y(\mathbf{u}_k) - \begin{pmatrix} 0 & s_f \end{pmatrix}^T\right)^2 \right) \\ &= -2 \sum_{\forall k} F_s(\mathbf{u}_k) \left(d_x^x(\mathbf{u}_k) - s_f + d_y^y(\mathbf{u}_k) - s_f\right) \\ s_f &= \frac{\sum_{\forall k} F_s(\mathbf{u}_k) \left(d_x^x(\mathbf{u}_k) + d_y^y(\mathbf{u}_k)\right)}{2 \sum_{\mathbf{u}_k} F_s(\mathbf{u}_k)}\end{aligned} \tag{4.18}$$

The optimal least squares solution to all constraints might include fold-overs of the warped pixel grid so that the output image is undefined in these regions. One approach [Wan+08] to address this problem is to increase the penalty for edge bending Eq. (4.5). However, this method cannot fully prevent fold-overs since the optimization might violate the edge bend constraint in favor of other energy terms. Moreover, this penalty introduces a global smoothing of the warp so that the available degrees of freedom cannot be utilized to retarget the image. We found that a more robust solution is to incorporate hard constraints (Section 3.3.2) with respect to the minimal allowed size ϵ

of a warped grid cell (i.e., pixel). In our current implementation we simply chose $\epsilon = 0.1$. This approach prevents fold-overs and has the considerable advantage that it does not introduce undesirable global smoothness into the warp (see Fig. 4.9). As a second advantage this size constraint prevents a complete collapse of homogeneous regions and other singularities in the warp which would result in visible artifacts.

Grid Scales and Iterations

Given the update steps for each free variable the multiscale optimization starts at the coarsest level where the corresponding update equations are derived from **A** and **b** using the so called full weighting approach [BHM00]. Due to the good convergence properties of our method the warp can be reinitialized in every frame based on the target scaling factors s_w and s_h. This considerably simplifies the construction of the multiscale hierarchy. In our current implementation the solver performs 40 iterations on coarse grid levels which are reduced to only 5 iterations at the pixel level resolution. For the free variables such as the uniform scale factor for feature regions s_f Eq. (4.2) or the line constraint parameters Eq. (4.13) optimized values are estimated after each iteration [Wan+08]. In Table 4.3 we provide timings and framerates for different input formats.

4.3.2 Rendering

EWA splatting of 3D surfaces can be performed efficiently on standard GPUs. See Section 3.5.3. Given that our solver is implemented in CUDA we can directly compute all necessary information on the GPU and save it in an OpenGL vertex buffer.

The undeformed pixel grid of an input frame I_t and corresponding splats representing the radial Gaussian basis functions Eq. (3.28) are illustrated in Fig. 3.2 (a). After computing the warp using our CUDA multigrid solver the warped splat positions $w_t(\mathbf{p})$ and the deformed splat shapes Fig. 3.2 (b), which are estimated from the corresponding Jacobian **J** as described in Section 3.5.3, are stored in an OpenGL vertex buffer.

4.4 Results and Comparison

In the this section we compare our method with previous work on image and video retargeting. In addition, we present an experimental evaluation in

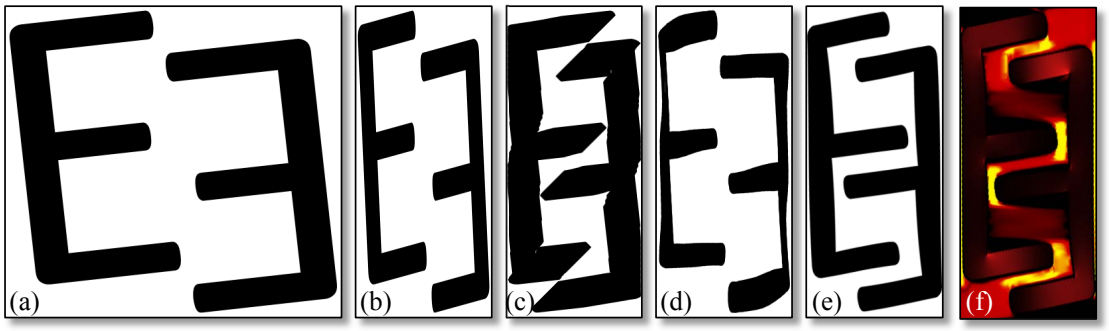


Figure 4.9: *Comparison to previous work. (a) Input frame. (b) Simple linear scaling. (c) Seam carving [RSA08]. (d) Optimized scale-and-stretch [Wan+08]. (e) Our method. (f) Illustration of the deformation energy.*

the form of a user study about the viewing preferences of 121 subjects. Key frame editing, additional comparisons, and examples are further illustrated in the accompanying video.

The instructional example of Fig. 4.9 demonstrates the benefit of our per-pixel warp compared to the seam carving method [RSA08] and to the optimized scale-and-stretch approach [Wan+08]. The 'E' shapes depicted in Fig. 4.9 (a) are marked as feature regions while the white background is marked as unimportant. The rescaled images have only 40% of the original width. Although seam carving generally preserves feature regions very well, it is limited by its iterative removal of seams with exactly one pixel per scanline. Hence it inevitably cuts diagonally through feature regions (Fig. 4.9 (c)). The optimized scale-and-stretch approach distributes the deformation more evenly, but it cannot scale feature regions uniformly due to the coarse grid and the missing per-pixel edge constraints (Fig. 4.9 (d)). Our per-pixel warp can fully utilize the available degrees of freedom to push the two shapes closer to each other while preserving their overall shape (Fig. 4.9 (e)). The corresponding deformation energy on the pixel grid is illustrated in Fig. 4.9 (f).

Similar effects can be observed in real-world images (Fig. 4.10). When rescaling the height down to 50%, seam carving is at first able to preserve most of the features. Yet, it eventually has to cut through feature regions to find a proper seam since it does not include any scaling (Fig. 4.10 (a)). The optimized scale-and-stretch approach emphasizes the center of the image and cannot bring the two persons closer together due to the coarse deformation grid, so that off-center features, such as the upper face, get distorted (Fig. 4.10 (b)). Our automatic retargeting preserves all feature regions equally well,

Figure 4.10: *(a) Seam carving [RSA08]. (b) Optimized scale-and-stretch [Wan+08]. (c) Our result.*

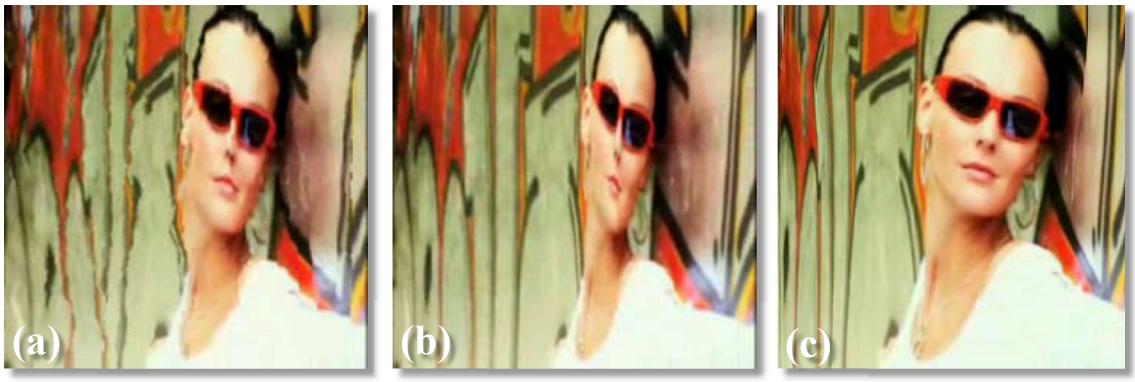


Figure 4.11: *(a) Seam carving [RSA08]. (b) [WGC07]. (c) Our result.*

and it retains relative proportions by distributing the deformation over the homogeneous regions in the background (Fig. 4.10 (c)). This example also illustrates the benefit of computing one single scale factor s_f for all feature regions Eq. (4.2).

A comparison of our method to the two current state-of-the-art methods for video retargeting, seam carving [RSA08] and the approach of [WGC07], is provided in Fig. 4.11. The example shows one of the main limitations of both methods, namely their inability to scale feature regions uniformly. Seam carving can only remove content and hence creates visible cuts. Similarly, the method of Wolf et al. produces visible discontinuities due to strong compression of image regions. The appearance of the main character is distorted in both cases.

Fig. 4.12 presents an additional comparison for the 3D animation movie 'Big Buck Bunny' and a soccer scene. Fig. 4.12 (a) shows the result of the seam carving approach, which again can only remove content, but does not allow for changes of scale. Our result is shown in Fig. 4.12 (b). Fig. 4.12 (c) and (d) compare linear scaling with a fully automatic video retargeting

Figure 4.12: *(a) Seam carving result for a frame from the movie Big Buck Bunny. (b) Our result. (c) Linear scaling of a soccer scene. (d) Our result.*

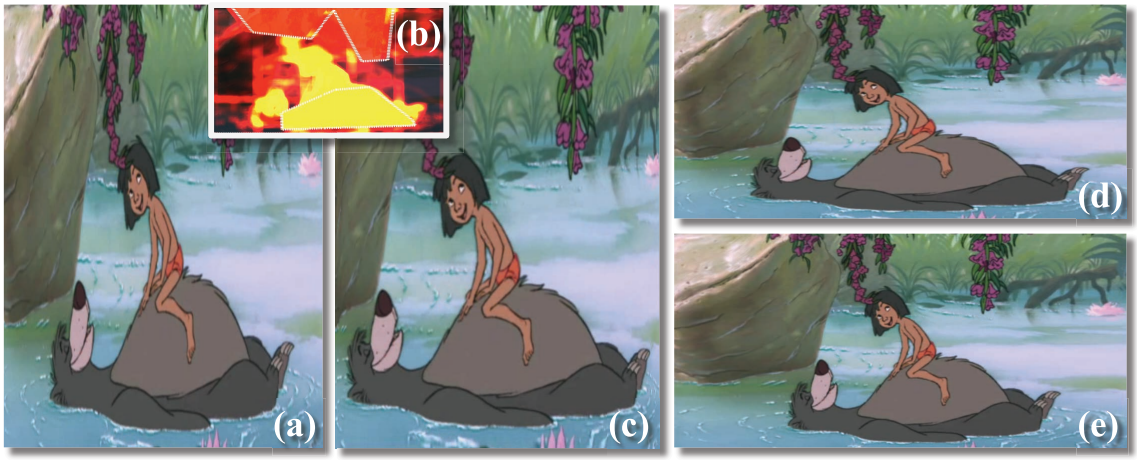


Figure 4.13: *(a), (d) Linear scaling. (b) Saliency. (c), (e) Our result.*

computed on close-up footage of a TV sports broadcast. As can be seen, the physical proportions of the players in Fig. 4.12 (d) appear much more realistic compared to the linear scaling. The same result is obtained for shots taken from the overview camera.

Interactive Constraint Annotation. For the Jungle Book example we rescaled the original video linearly down to 50% separately along the x-axis (Fig. 4.13 (a)) and the y-axis (Fig. 4.13 (d)). In general, automatic saliency estimation is difficult for 2D cartoons because characters, such as Mowgli and Baloo, are drawn by large homogenous regions while the background artwork exhibits much more complex structure. For this scene we applied a simple manual annotation to the saliency map (Fig. 4.13 (b)). It emphasizes the characters and reduces the importance of the background. As shown in Fig. 4.13 (c) and (e) this single modification retargets the video faithfully to considerably different aspect ratios such as those occurring when reformatting from wide screen to DVD.

Figure 4.14: *(a) Input image of a house. (b) Automatic result. (c) Added position constraint. (d) Line constraint for the fence.*

Figure 4.15: *(a) Automatic rescaling of a seesaw image. (b) With two added line constraints.*

Fig. 4.14 (a) shows a house scene which has been rescaled to 50% of the original width in Fig. 4.14 (b). The automatic saliency detection classifies the sky as unimportant so that this region is overly enlarged by our warp. In order to achieve a more balanced visual appearance the user adds an additional positional constraint for the house in Fig. 4.14 (c). The unnatural deformation of the fence can be eliminated by adding a single line constraint (Fig. 4.14 (d)). Automatic retargeting of an image of a seesaw to 50% of the original height does not preserve the straight bars (see Fig. 4.15 (a)). Such problems may arise in cases where the automatic saliency estimation is difficult due to prevalent global images structures. However, by adding two line constraints as in Fig. 4.15 (b) the bending problem is resolved. An additional example is shown in Fig. 4.4.

As mentioned in Section 4.2.3 most results are based on a default parameter set. For some examples like fast-paced sport scenes it is beneficial to reduce, e.g., the weight of the temporal coherence to let the warp better adapt to fast player and camera movements. For animation movies and cartoons, which often have dominant silhouettes, we increased the weights for edge bending and edge sharpness. Due to our real-time pipeline the effect of changing these parameters can be intuitively explored by the user. The weight presets used

Images (c)-(f) ©Disney

Figure 4.16: *Limitations. (a) Linear scaling of an image with strong structure. (b) Our result. (c), (e) Linear scaling of video with very dynamic motion and rapid camera movement. (d), (f) Our result.*

Table 4.1: *Weight presets for different scene types.*

Scene type	λ_b	λ_s	λ_c
Default	100	10	10
Animation movie	110	20	10
Sport	110	10	1
Text	100	70	10

for our results are provided in Table 4.1. A demonstration of the parameter sensitivity is shown in the accompanying video.

4.5 User Study

Despite the discussed technical advantages of our method, the most important criterion for the utility of a video retargeting method is whether it is actually preferred by the viewer. Hence we conducted an experimental evaluation in the form of a user study with 121 participants of different age, gender, and education to evaluate viewing preferences regarding the current state-of-the-art techniques for video retargeting. One of the most suitable standard methods for statistical evaluation of subjective preferences is the method of *paired comparisons* [Dav63]. In this method, items are presented side-by-side in pairs to an observer, who then records a preference for one of the members of the pair. Following this aproach, we prepared an online survey showing

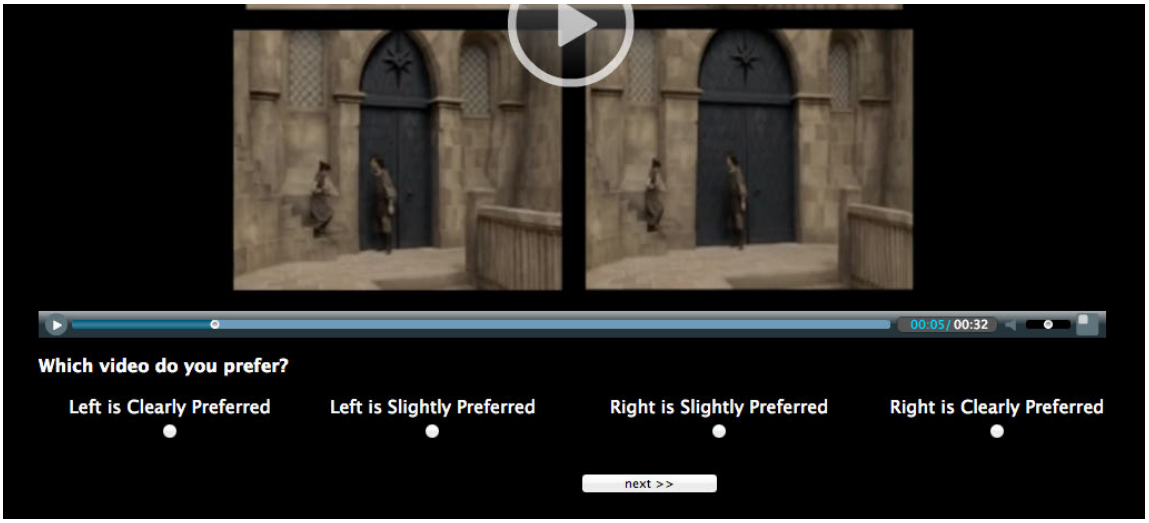


Figure 4.17: *This shows a screen shot of the web-based user study we have implemented. We asked 121 people to rate the retargeting quality for 6 different videos. We compared three retargeting techniques and uniform scaling. We used pairwise comparison.*

pairs of retargeted video sequences. For each pair the viewer simply had to pick the preferred video. We compared automatically generated results of our method (using the default parameters and no user editing) to the methods of [RSA08] and [WGC07] for six input videos. Hence the survey consisted of 18 video pairs and we received $18 \times 121 = 2178$ answers overall. Each individual method was compared $2 \times 6 \times 121 = 1452$ times. We tried to minimize bias, e.g., by randomizing the order of pairs and by providing only the most necessary information, without technical details, to the participants, since drawing attention to particular artifacts might influence the actual viewing preferences.

Table 4.2: *Preferences of 121 persons for 3 retargeting techniques. For example, an entry n in row 1 and column 2 means that the result of method 1 was preferred n-times to the result of method 2.*

	1	2	3	Total (2178)
1. Our method	-	553	559	1112
2. [WGC07]	173	-	449	622
3. [RSA08]	167	277	-	444

Table 4.2 shows how many times the result of a particular method was preferred by the participants. The resulting ranking shows a clear preference

for our method. Our results were favored in 76.2% (553 of 726) of the comparisons with Wolf et al. and in 77% (559 of 726) of the comparisons with Rubinstein et al. Overall, the participants favored our method in 76.6% (1112 of 1452) of the cases. Methods 2 and 3 were preferred in 42.8% (622 of 1452) and 30.6% (444 of 1452) of the comparisons with the respective other two methods. The intraobserver variability, Kendall's coefficient of consistence $\zeta \in [0,1]$, had a very high average of $\bar{\zeta} = 0.96$ and a small standard deviation $\sigma = 0.078$. This indicates that each single participant had clear preferences without substantial inconsistencies (i.e., circular triads like $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$). 80.9% of the participants had perfectly consistent preferences with $\zeta = 1$. Only two subjects had a value of $\zeta = 0.66$. This, however, means that they still had consistent preferences for 4 of the 6 videos. The interobserver variability, Kendall's coefficient of agreement, is $u = 0.206$ for Table 4.2, with a p-value < 0.01. Hence, there is a statistically significant agreement among the participants regarding the three methods. We refer to [Dav63] for a detailed explanation of these indicators.

A pairwise comparison including linear scaling would have required each participant to select 36 video preferences instead of 18. Since this would have been a tedious procedure, we instead asked the participants to rank the three methods and a linearly scaled version for each of the six input videos (i.e., 726 rankings of the four methods) from 1 (most preferred) to 4 (least preferred). The average ranks were: our method 1.66, [WGC07] 2.49, linear scaling 2.73,[**c**]iteRubinstein2008 3.12. This result confirms the preferences in Table 4.2 and also indicates that our retargeted video is generally preferred over linear scaling. This is an important observation regarding the general utility of video retargeting.

Real-time Performance. Performance figures of our method for different input formats are provided in Table 4.3. The reference system was a 2GHz AMD Dual Core CPU with 2GB of memory and a single NVIDIA GTX280 graphics adapter. We break down timings for the main computational steps such as feature estimation, multigrid optimization, and EWA splatting. The total figures include additional processing steps like the streaming of video frames to the GPU. Our method achieves frame rates of over 20 FPS at NTSC resolution and still works at interactive rates with approximately 10 FPS for HDTV resolutions. Furthermore, the performance is largely independent of the output resolution.

Limitations. Prominent spatial and temporal elements like buildings or complex motions without sufficient homogenous regions to absorb the deformation pose a fundamental limitation to any type of non-linear image resizing. In these cases the warp does not have sufficient degrees of free-

Table 4.3: *Per-frame times (ms) and FPS for different input formats.*

Input	Features	Opt.	EWA	Total	FPS
320×180	5.6	9.2	3.2	21.1	47.4
480×270	7.5	13.5	4.0	29.8	33.5
640×480	12.3	22.5	6.6	45.9	21.8
720×384	11.2	21.3	5.9	43.2	23.1
1280×720	27.6	48.3	11.1	102.4	9.7

dom to compress regions without violating feature constraints. Our warp automatically falls back to linear scaling in these situations (Fig. 4.16). We believe that this is a positive property, since it does not introduce too many undesirable non-linear deformations for this type of input. In some cases, where the automatic saliency computation detects large salient regions, our method (similar to previous work) tends to compress content at the image boundary. In our system, this can be resolved by our manual warp constraints. However, we think that a combination with retargeting operators like cropping or zooming might also provide improved, automatically generated results [RSA09]. Our current sliding window approach to handle temporal coherence was motivated by our aim to process video in real-time. Preprocessing the full video allows to keep the distortion constant across the optical flow which results in improved temporal coherence for complex motion [Wan+09b]. Fortunately, such a pre-analysis could be easily integrated into our post-production pipeline by storing and streaming the corresponding high level temporal constraints in form of additional annotations with the video.

4.6 Conclusion and Future Work

In this chapter we have proposed a system for video retargeting that builds up on our novel image wapring and rendering framework. We introduced some application specific additional conceptual as well as technical novelties. The simple but powerful interactive framework surrounding the image warping combines a variety of automatic constraints with interactive annotations of streaming video. This enables content producers to add high level constraints with respect to scene composition or artistic intent. These constraints remain valid across different target formats and hence allow for an art directable retargeting process. Our major technical contributions include various improvements and extensions of automatic constraints, such as

bilateral temporal coherence. In contrast to previous retargeting approaches our framework allows to compute the warp at the pixel resolution and we achieve aliasing reduced high quality results with our 2D EWA rendering technology. A user study revealed a clear viewer preference for the results of our method over previous approaches and linear scaling.

Our key frame based constraint annotation has been designed according to common practice in standard video editing tools, and we received encouraging feedback from various companies focusing on video production. However, there is certainly room for improvement on our interaction methods. Nevertheless, our approach demonstrates that future practical solutions will have to be semi-automatic. It is the combination of high level, interactive control over scene composition with low level automatic feature detection that stands as a key requirement for production environments.

Besides addressing the limitations mentioned above, we would like to extend our system in several respects. For example, in some application domains certain high level constraints could be provided automatically, like line markings on the pitch for soccer or rescaling constraints for 3D animation movies. Finally, higher level perceptual metrics and more detailed studies should be used to assess the quality of the warp and to compare different methods.

After we presented this video retargeting project at Siggraph Asia 2009 conference we could happily see many follow-up projects based on our technologies. We saw third party user studies which compared to even more video retargeting technologies and still reported our method as one of the top performing ones([Rub+10]). Also there is large body of work the refine some aspects of our method. In particularly, [Gre13] implemented our retargeting application on silicon hardware, such that it could potentially much easier integrated in consumer electronics products.

In the next chapter we will extend the insights we gained from the video retargeting to the field of stereoscopic video. Instead of one video stream we will handle two or more streams simultaneously and introduce constraints that span over all videos and allow not only to retarget video two-dimensionally to a new aspect ratio but also allowing us to retarget the entire perceived 3D space to better target a given stereoscopic output device.

CHAPTER 5

Disparity Mapping

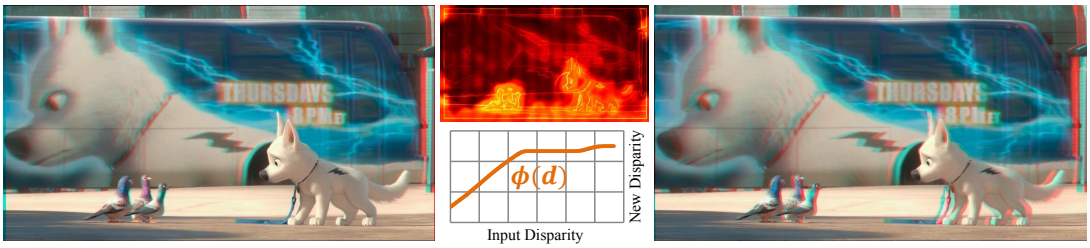


Figure 5.1: *Our method retargets stereoscopic 3D video automatically to a novel disparity range, based on visual importance of scene elements and a nonlinear disparity mapping operator ϕ. This retargeting is accomplished using a novel stereoscopic image warping technique.*

In this chapter we extend our general video warping and rendering framework to the field of stereoscopic video manipulation. We draw upon insights gained from the video retargeting application to implement a pipeline that is capable of warping multiple video streams simultaneously. This allows us to formulate warp constraints that span over many input video streams. We will show that besides allowing us to simply extend the aspect ratio change application to be consistent in a multi-view setup, we can introduce a much more important novel way of changing the disparity between input views. In this chapter we will show that such disparity change is highly important

because it can addresses the problem of remapping the disparity range of stereoscopic images and video. Such operations are highly important for a variety of issues arising from the production, live broadcast, and consumption of 3D content.

Our work is motivated by the observation that the displayed depth and the resulting 3D viewing experience are dictated by a complex combination of perceptual, technological, and artistic constraints. We first discuss the most important perceptual aspects of stereo vision and their implications for stereoscopic content creation. We then formalize these insights into a set of basic *disparity mapping operators*. These operators enable us to control and retarget the depth of a stereoscopic scene in a nonlinear and locally adaptive fashion. For fields other than stereoscopic 3D video, this content *retargeting* or *remapping* problem has been investigated extensively in computer graphics. For example, tone mapping techniques [Rei+05] exploit properties of our color perception to adapt HDR images to display devices of lower dynamic range and vice versa by nonlinear remapping of colors. We introduce similar mapping concepts to field of stereoscopic content production and display.

From a sparse set of stereo correspondences, our algorithm computes disparity and image-based saliency estimates, and uses them to compute a deformation of the input views so as to meet the target disparities. Our approach represents a practical solution for actual stereo production and display that does not require camera calibration, accurate dense depth maps, occlusion handling, or inpainting. We demonstrate the performance and versatility of our method using examples from live action postproduction, 3D display size adaptation, and live broadcast. An additional user study and ground truth comparison further provide evidence for the quality and practical relevance of the presented work.

5.1 Stereoscopic Disparity

As motivated in the introduction Section 1.2, stereoscopic 3D production and display is a complex field, involving a broad range of research and experience on human visual perception [HR02], display technology [Hof+08], and industrial best practice [Men09]. Addressing all involved issues requires multi-disciplinary research efforts where progress in one field might lead to the need for new research in related fields. However, similar to the now well established fields of media retargeting or tone mapping, there exist a number of fundamental insights about stereoscopic perception and display, which are of highest relevance in application domains such as 3DTV and 3D cinema.

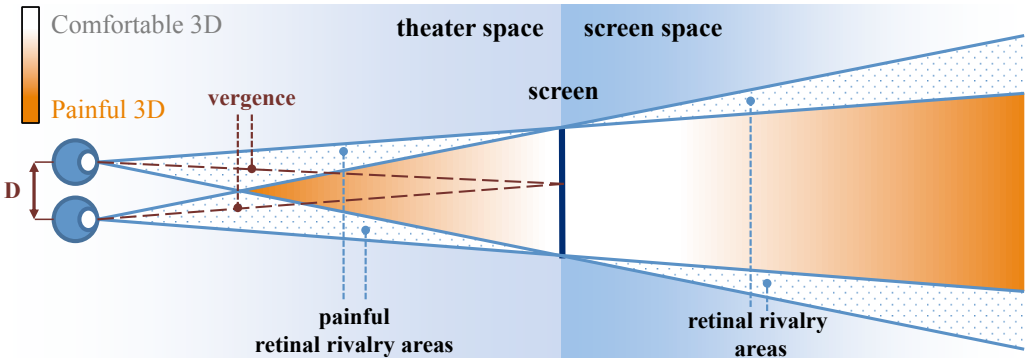


Figure 5.2: *Illustration of the stereoscopic comfort zone.*

One of the central parameters in stereoscopy is *disparity*. This section is concerned with the discussion of four of the most important problems related to disparity [HR02; Men09]. We will provide an overview of these issues from a perceptual point of view, and describe how they are addressed in today's stereoscopic production pipeline in Section 5.1.1. Based on these basic rules and guidelines, we then propose a set of *disparity mapping operators* in Section 5.1.2 which formalize these ideas and provide a first basic and extendable framework for general disparity editing of stereoscopic 3D footage.

5.1.1 Background from Stereography

Disparity Range

Our visual system has several constraints regarding the admissible distance of corresponding points on the retina that still allows for proper depth perception. The central parameters influencing retinal disparity are the *interocular distance*, the *vergence* of the eyes, and the distance to the point of interest. For example, if we focus on a nearby object, the images of other objects in the distant background cannot be fused by our visual system anymore due to too large retinal disparities and will appear as double images (please refer to [HR02] for more detail). Depending on the above parameters, there is only a restricted disparity range around the *Horopter*, called Panum's area, which permits proper stereo vision and depth perception.

A central challenge in stereoscopic movie production is that current display technologies only have indirect control over these parameters, which they achieve by presenting a pair of slightly different images to the left and right eyes. The only parameter, which can be directly controlled, is the distance between corresponding features in the two displayed images, the *image disparity*. The actual retinal disparity then results from the convergence

of the eyes, distance to and size of the screen, etc. In the following, when we discuss changing the "disparity" of a scene, we refer to the modification of the image disparity. A further problem is that important depth cues such as accommodation cannot be controlled at all; we have to focus our eyes on the screen surface, even if objects are positioned in front or behind the screen surface, resulting in conflicting depth cues. These technological limitations can lead to considerable problems, ranging from distortions of the 3D structure of a scene to visual fatigue [Hof+08]. With existing capture and display technologies these fundamental problems cannot be resolved. Hence, the disparity range for a displayed scene has to be adapted in order to minimize these issues.

In production, the admissible disparity range is the so called *comfort zone* (see Fig. 5.2). Modifying stereoscopic content and disparity ranges to a generic comfort zone suitable for large population groups has been investigated, for example, in the context of Microstereopsis [SN00]. A prominent solution today is *linear* disparity remapping [SH09; Men09]. Such a linear mapping changes the disparity interval from a given range to the desired range. The introduced linear distortion of the disparity space amounts to uniform flattening of objects in the scene. Some concrete applications for linear disparity range mapping are the adaptation of stereoscopic content to display devices of different size, the uniform compression of scene depth for a more comfortable viewing experience, or moving scene content forward or backward by adding a constant offset to the disparity range. Practical values for disparity on a 30 foot cinema screen, are between +30 (appears behind screen) and -100 (appears in front of screen) pixels, assuming video with a width of 2048 pixels. In practice such a mapping can be achieved by modifying the camera baseline (the interaxial distance) during filming, and by shifting the relative position of the left and right view after filming to control the absolute disparity offset. For instance, objects that are floating in front of the screen and intersect with the image borders will cause retinal rivalry (see Fig. 5.2). In post-production this can be corrected using the floating window technique, which is a virtual shift of the screen plane towards the viewer [Men09]. In general, however, such adaptations have to be performed by expensive and cumbersome manual per-frame editing, since the camera baseline of recorded footage cannot be easily modified.

These disparity range limitations are the most obvious issue in stereoscopic perception and production. However, related to this limitation are a number of further issues, which we shall describe.

Disparity Sensitivity

Our ability to discriminate different depths decreases with increased viewing distance. One result from perceptual research is that the stereoacuity is inversely proportional to the square of the viewing distance [HR02]. This means that our depth perception is generally more sensitive and accurate with respect to nearby objects, while for distant objects other depth cues such as occlusion or motion parallax are more important [BGL04].

This effect can be exploited in stereoscopic movie production by compressing the disparity values of distant objects. For example, a disadvantage of the previously mentioned linear range adaptation is that strong disparity range reduction leads to apparent flattening of objects in the foreground. Using the insights about stereoacuity, the decreased sensitivity to larger depths can be used to apply *nonlinear* remapping instead, resulting in less flattening of foreground objects. Effectively, this corresponds to a compression of the depth space at larger distances. This idea can be extended to composite nonlinear mapping, where the disparity range of single objects is stretched, while the space in between the objects is compressed. Such nonlinear operations which exploit the limitations in sensitivity of our visual system have been successfully employed in related areas such as media retargeting. But so far, they are difficult to apply to stereoscopic footage of live action, since this would require an adaptive modification of the camera baseline. In production, the only way to achieve such effects is complex multi-rigging by capturing a scene with camera rigs of varying baseline and manual composition in post-production [Men09].

Disparity Gradient

Besides limitations with respect to absolute disparity values, experiments have shown that our perception is subject to limits regarding the disparity gradients in an observed scene as well [BJ80]. In particular, the perception of different depth gradients strongly depends on local scene content, spatial relationships between objects, etc. Consequences range from distorted perception of the 3D structure of a scene to the inability to see proper stereo.

Exploiting the different types of gradient sensitivities of our visual perception (e.g., regarding color) has proven to be a valuable tool in research on tone mapping, where locally adaptive gradient domain processing is used for content-aware color remapping. In stereography, locally adaptive disparity modifications are important in two respects. On the one hand, it has to be ensured that the displayed gradients do not violate our perceptual limits. On

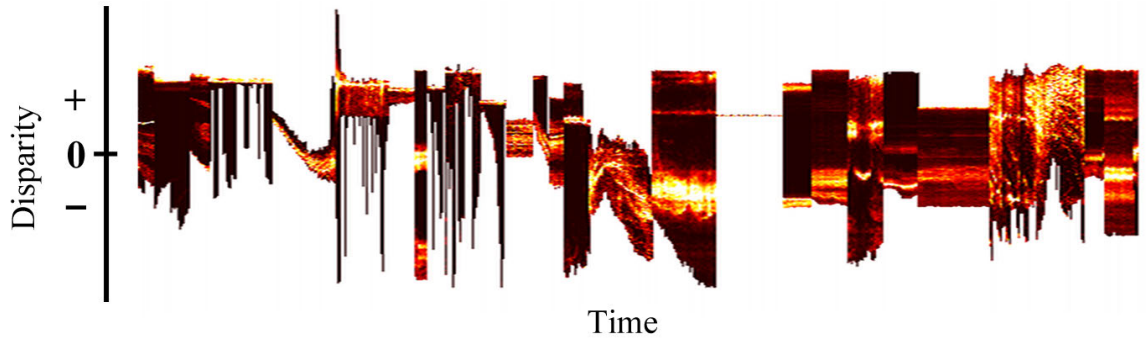


Figure 5.3: *A disparity storyboard for a 3D movie. In this plot, the range of estimated disparity values is plotted on the vertical axis on a frame-by-frame basis. The color indicates the frequency of the occurrence of disparity values. The velocity is visible in the changes in disparity histograms over time.*

the other, disparity gradient editing provides the possibility to redesign the depth structure of a scene on an object basis. This type of artistic freedom is a highly desired feature in post-production, but extremely difficult to achieve at the moment (e.g., using the previously mentioned multi-rigging).

Disparity Velocity

The last important area is the temporal aspect of disparity. For real world scenes without conflicting stereo cues, it has been shown that our visual system can rapidly perceive and process stereoscopic information. The reaction time, however, can increase considerably for conflicting or ambiguous cues, such as inconsistent vergence and accommodation. Moreover, there is an upper limit to the temporal modulation frequency of disparity [HR02].

These temporal properties have considerable importance in the production of stereoscopic content. In the real world we are used to disparities varying smoothly over time. In stereoscopic movies, however, transitions and scene cuts are required. Due to the above mentioned limitations such strong discontinuities are perceptually uncomfortable and might again result in the inability to perceive depth [Men09]. Therefore, stereoscopic film makers often employ a continuous modification and adaption of the depth range at scene cuts in order to provide smooth disparity velocities, so that the salient scene elements are at similar depths over the transition. Additionally, such depth discontinuities can be exploited explicitly as a storytelling element or visual effect and are an important tool used to evoke emotional response. Fig. 5.3 illustrates disparity histograms of a 3D movie over time, where such smooth

transitions are visible. These disparity storyboards are an important part of the planning required for 3D productions.

We summarize the four central aspects of disparity which we utilize to design the disparity mapping operators in the following section:

Disparity Range: Mapping of the global range of disparities, e.g., for display adaptation.
Disparity Sensitivity: Disparity mapping for global or locally adaptive depth compression and expansion.
Disparity Gradient: Content-adaptive disparity remapping by modifying disparity gradients.
Disparity Velocity: Temporal interpolation or "smoothing" between different disparity ranges at scene transitions.

These operations are of essential importance for the generation and display of 3D footage, be it during post-production of movies or real-time correction of live broadcasts. In the following section we will formalize these insights into corresponding disparity mapping operators and then present a novel framework that allows us to perform complex disparity editing on existing stereo footage.

5.1.2 Disparity Mapping Operators

We will first consider disparity mapping operators on a conceptual level. Section 5.3 will then provide examples for relevant application scenarios. Without loss of generality we assume that the input footage is recorded with a stereo camera rig or is approximately rectified. For a digital stereo image pair (I_l, I_r) let $\mathbf{u} \in \mathbb{R}^2$ be a pixel position in the left image I_l. We define the disparity $\sigma(\mathbf{u}) \in \mathbb{R}$ as the distance (measured in pixels) to the corresponding pixel in I_r (and vice versa). The range of disparities between the two images is an interval $\Omega = [\sigma_{min}, \sigma_{max}] \subset \mathbb{R}$. Our disparity mapping operators will be defined as functions $\phi : \Omega \to \Omega'$ which implement the rules and guidelines described in the previous section by mapping an original range Ω to a new range Ω'. For illustration we refer to the examples in Section 5.3.

Linear Operator

Globally linear adaptation of a disparity $\sigma \in \Omega$ to a target range $\Omega' = [\sigma'_{min}, \sigma'_{max}]$ can be obtained by a mapping function

$$\phi_l(\sigma) = \frac{\sigma'_{max} - \sigma'_{min}}{\sigma_{max} - \sigma_{min}}(\sigma - \sigma_{min}) + \sigma'_{min}. \tag{5.1}$$

By changing the interval width of Ω', the depth range can be scaled and offset such that it matches the overall available depth budget of the comfort zone (e.g., Section 5.3 and Fig. 5.13).

Nonlinear Operator

Global nonlinear disparity compression can be achieved by any nonlinear function, e.g.,

$$\phi_n(\sigma) = \log(1 + s\sigma), \tag{5.2}$$

with a suitable scale factor s. For more complex, locally adaptive nonlinear editing, the overall mapping function can be composed from basic operators. For example, given a set of different target ranges $\Omega_1, \ldots, \Omega_n$ and corresponding functions $\phi_0, \ldots, \phi_n$, the target operator would be:

$$\phi_a(\sigma) = \begin{cases} \phi_0(\sigma), & \sigma \in \Omega_0 \\ \quad \ldots & \quad \ldots \\ \phi_n(\sigma), & \sigma \in \Omega_n \end{cases}. \tag{5.3}$$

An elegant approach to generate such complex nonlinear functions in a depth authoring system is to either use the histogram of disparity values (as shown in Figures 5.3 and 5.4) for identifying dominant depth regions, or to analyze the visual saliency of scene content in image space. These so called saliency maps $F_s(\mathbf{u}) \in [0,1]$ (see Fig. 5.5) represent the level of visual importance of each pixel and can be generated either automatically by the system (see also Section 5.2.2) or manually by the user. From the saliency map, the algorithm can infer which disparity ranges Ω_i are occupied by important objects, and which regions are less important. From these importance values, which essentially correspond to the first derivative ϕ'_a, the actual disparity operator can be generated as the integral $\phi_a(\sigma) = \int_0^\sigma \phi'_a(u)du$ (please see Figures 5.1 and 5.7 for examples).

Gradient Domain Operator

In addition to local adaptivity in disparity space as in ϕ_a, the remapping of disparity gradients allows for additional spatial adaptivity in image space. Retargeting operators for disparity gradients with spatial adaptivity can be defined based on visual importance maps $F_s(\mathbf{u})$ as functions $\phi_\nabla(\nabla\sigma(\mathbf{u}), F_s(\mathbf{u}))$. An example for adaptive compression using interpolation between a linear and a nonlinear map ϕ_l and ϕ_n is

$$\begin{aligned} \phi_\nabla(\nabla\sigma(\mathbf{u}), F_s(\mathbf{u})) = & F_s(\mathbf{u})\phi_l(\nabla\sigma(\mathbf{u})) + \\ & (1 - S(\mathbf{u}))\phi_n(\nabla\sigma(\mathbf{u})). \end{aligned} \tag{5.4}$$

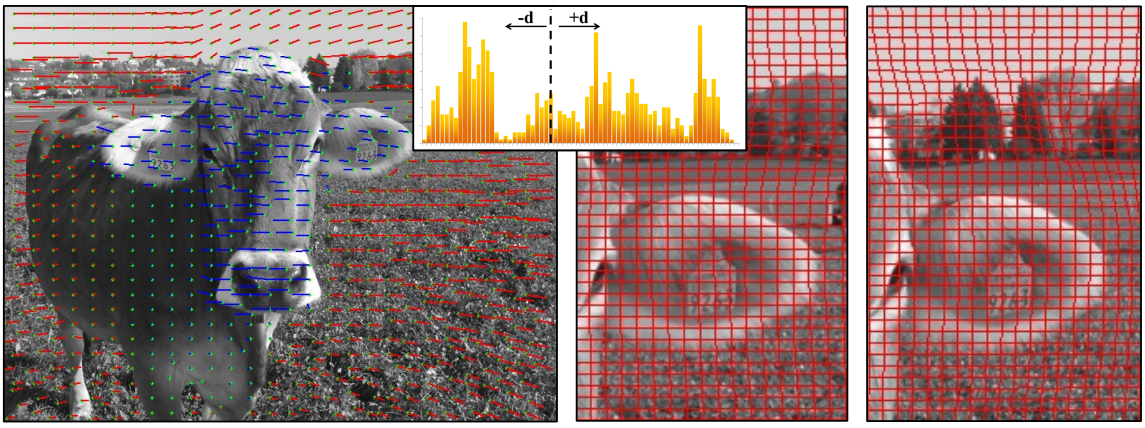


Figure 5.4: *Left: stereo correspondences with color coded disparities (red positive, blue negative) and the disparity histogram for the cow example after pruning. Right: close-ups of the warped stereo pair showing the deformed isolines with respect to the input views.*

The actual disparity mapping operator can then be reconstructed from ϕ_∇ using methods from gradient domain processing [AR07].

Temporal Operator

Temporal adaptation and smoothing, as it is required for smooth scene transitions or visual effects, can be defined by weighted interpolation of two or more of the previously introduced operators, e.g.,

$$\phi_t(\sigma, t) = \sum_i w_i(t)\phi_i(\sigma), \tag{5.5}$$

where $w_i(t)$ is a suitable weighting function. An example for temporal interpolation and the resulting disparity histograms is given in Section 5.3, Fig. 5.9.

These different operators in Eq. (5.1)-(5.5) provide the basic functionality required to implement the set of central disparity operations presented in Section 5.1.1. In the following section, we will present our novel image-based stereoscopic warping scheme for applying these operators to stereo footage. Section 5.3 will then provide concrete applications for these operators in production.

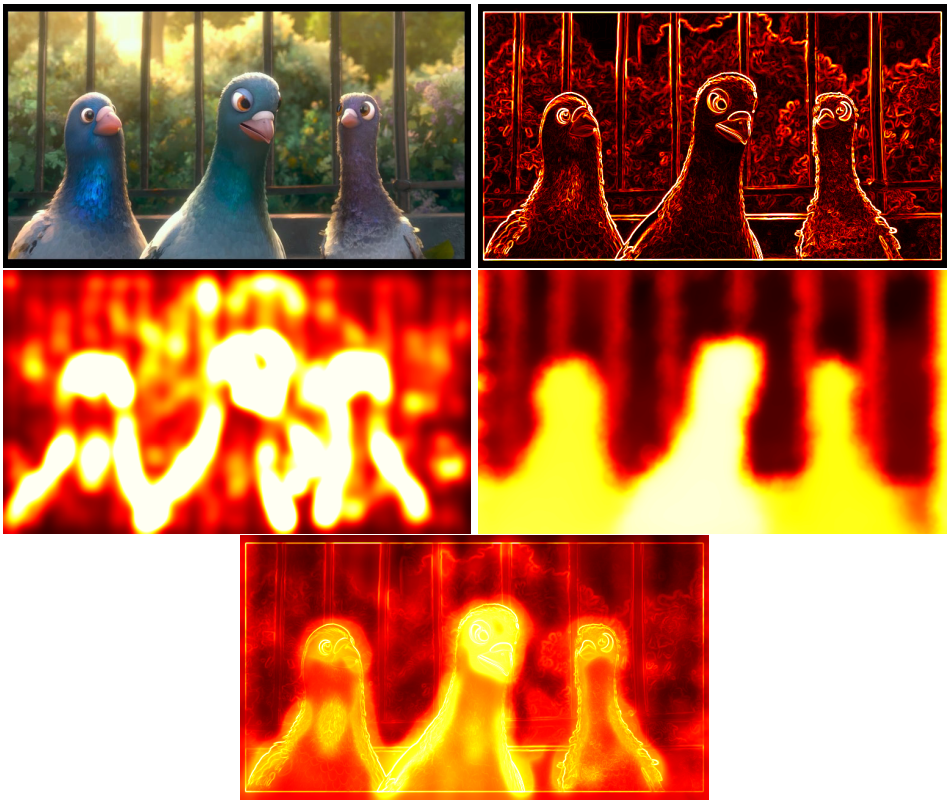


Figure 5.5: *Individual saliency components and final automatically generated saliency map for a scene. From left to right: left image of a stereo pair, local edge saliency, global texture saliency, disparity-based saliency, combined saliency map F_s.*

5.2 Stereoscopic Warping

5.2.1 Sparse Stereo Correspondences

Sparse feature correspondences between the two images (I_l, I_r) can be estimated robustly using well established standard techniques [BM04; Low04; Sin+06], hence we refer to these works for detail on the basic correspondence matching. Optionally we exploit downsampled dense correspondence information [Wer+09] between I_l and I_r for large textureless image regions which are too ambiguous for sparse feature matching. Outliers can be removed automatically [SLK09]. In Chapter 6 we introduce a general framework that can create high-quality disparity maps. In can directly be used to obtain correspondences for the disparity mapping operator.

Depending on scene content the resulting feature set $\mathcal{F}$ generally has an irregularly clustered distribution of correspondences. Moreover, many fea-

tures are not temporally stable over a longer video sequence, but disappear after a few frames. Since our warping algorithm requires only a sparse set of features, we apply a spatially anisotropic pruning algorithm to $\mathcal{F}$ which favors temporally stable correspondences and is adaptive to depth discontinuities. Correspondences are first sorted by their lifetime, so that long living pairs receive a high priority and correspondences which appeared only for a couple of frames receive a low priority. We then apply a greedy procedure to remove low priority correspondences around those with a high priority. Let $(\mathbf{u}_l, \mathbf{u}_r) \in \mathcal{F}$ be a high priority correspondence pair with disparity $\sigma(\mathbf{u}_l)$. Our pruning algorithm removes all pairs $(\mathbf{u}'_l, \mathbf{u}'_r) \in \mathcal{F}$ with

$$\left\| \begin{pmatrix} \mathbf{u}_l \\ \sigma(\mathbf{u}_l) \end{pmatrix} - \begin{pmatrix} \mathbf{u}'_l \\ \sigma(\mathbf{u}'_l) \end{pmatrix} \right\| < r. \tag{5.6}$$

This isotropic distance measure in image and disparity space results in a locally adaptive anisotropic filter in image space only (similar to the idea of the Fast Bilateral Filter [PD06]). Pruning is performed symmetrically for the positions $\mathbf{u}_r$ as well. In principle, the radius r depends on the image resolution and the disparity range. However, the results of our warping algorithm are quite insensitive to different feature densities so that we could simply use a value of $r = 10$ in our experiments.

Fig. 5.4 shows an example of the resulting features and the corresponding disparity histogram. This algorithm combines the respective strengths of different methods for feature estimation and provides a robust way to automatically compute a sparse but sufficiently accurate set $\mathcal{F}$ of correspondences between stereo pairs.

5.2.2 Depth and Image Saliency

In order to determine which parts of the input images can be distorted by our warp without creating visible artifacts, we need a visual importance map F_s for the stereo image pair. Our approach to compute f_s is twofold. First, we use image-based importance measures which are able to capture the coarse and fine scale details of image content, such as prevalent edges or textured regions. This is identical to the salience computation for video retargeting in Section 4.2.1. In addition, we have the sparse disparity information from the previously computed stereo correspondences. This allows us to exploit the depth dimension as an additional source of information to estimate visual saliency. Accordingly, we compute a composite saliency map as a weighted combination

$$F_s(\mathbf{u}) = \lambda F_{s,i}(\mathbf{u}) + (1 - \lambda) F_{s,\sigma}(\mathbf{u}), \tag{5.7}$$

for all pixels $\mathbf{u} \in I_l$ where $F_{s,i}$ represents the image-based saliency and $F_{s,\sigma}$ our disparity-based saliency. Similar to Section 4.2.1, $F_{s,i}$ is generated from the sum of a local edge map and the global scale method of Guo et al. [GMZ08] (see Fig. 5.5) for each stereo channel individually.

The disparity saliency map $F_{s,\sigma}$ can be computed by any operator on the range of disparities of correspondences in $\mathcal{F}$. A simple but effective solution is to assume that foreground objects generally catch our visual attention more than the background of a scene, which is a reasonable assumption for many application scenarios (see Section 5.1.1). So for a correspondence set $\mathcal{F}$ comprising a disparity range $\Omega = [\sigma_{min}, \sigma_{max}]$, we assign high saliency values to disparities close to σ_{min} and a low saliency to disparities close to σ_{max}. Saliency values are then interpolated over the non-feature pixels (see Fig. 5.5). Please see Chapter 6 where we introduce a novel method for sparse to dense interpolation over an image. Note that in principle more complex disparity-based saliency estimators are possible (see also Section 5.3).

Fig. 5.5 shows all components of the saliency computation. Dark areas in the final map are parts of the scene that are more likely to be distorted by the warp to accommodate movement within the images. For weighting S_i and S_d our current implementation uses a value $\lambda = 0.5$.

5.2.3 Warping

Our aim is now to warp the stereo image pair (I_l, I_r) such that the range of disparities Ω of the stereo correspondences $\mathcal{F}$ is mapped to a new range defined by a disparity mapping operator $\phi : \Omega \rightarrow \Omega'$. This means we have to compute a pair of warping functions (w_l, w_r) which map coordinates from (I_l, I_r) to a pair of output images (O_l, O_r). Note that in principle warping only a single image would be sufficient. However, by distributing the required deformation to both images, we are more flexible regarding the admissible disparity mapping operations without introducing noticeable visual artifacts. To compute these warps we employ again our novel image warping framework introduced in Chapter 3. We define a set of constraints on the functions (w_l, w_r) which can then be solved as a nonlinear least-squares energy minimization problem. The main difference now is that we construct one optimization problem involving all degree of freedom of both video streams. We have two warp grids, one for each frame, but all the grid vertex positions are then put in one optimization. This allows as to specify warp constraint the link grid vertices of the left and right image together.

5.2.4 Stereoscopic Constraints

The most central set of constraints applies the disparity mapping operator ϕ to the stereo correspondences $(\mathbf{u}_l, \mathbf{u}_r) \in \mathcal{F}$. For each correspondence pair we require

$$w_l(\mathbf{u}_l) - w_r(\mathbf{u}_r) - \phi(\sigma(\mathbf{u}_l)) = 0, \tag{5.8}$$

meaning that the disparity of a warped correspondence pair $(w_l(\mathbf{u}_l), w_r(\mathbf{u}_l))$ should be identical to applying the disparity mapping operator ϕ to the original disparity $\sigma(\mathbf{u}_l)$.

Since the above constraints only prescribe relative positions, we require a small set of absolute position constraints which fix the global location of the warped images. In the video retargeting application such absolute constraints where used only at the image edges (edge vertices were set to the new target aspect ratio Section 4.3.1). However, for the disparity mapping algorithm it is better not to restrict the edge pixels and therefore allow images to shift horizontally. But this requires another set of border constraints. For that we compute absolute position constraints for the 20% temporally most stable feature correspondences, i.e., those features which have been detected throughout a sequence of frames in the video. The warped positions are defined by the average previous position and the novel disparity:

$$\begin{aligned} w_l(\mathbf{u}_l) &= \frac{\mathbf{u}_l + \mathbf{u}_r}{2} + \frac{\phi(\sigma(\mathbf{u}_l))}{2} \\ w_r(\mathbf{u}_r) &= \frac{\mathbf{u}_l + \mathbf{u}_r}{2} - \frac{\phi(\sigma(\mathbf{u}_l))}{2} \end{aligned} \tag{5.9}$$

Eq. (5.8) and (5.9) define the basic stereoscopic warping constraints so that the warped images match the target disparity range Ω'.

5.2.5 Temporal Constraints.

For video with moving scene elements one has to ensure that local image distortion is properly transferred along the local motion flow [Wer+09] between successive video frames. The local image distortion can be measured based on the derivatives of the warp. Let $\partial w_x^t / \partial x$ denote the partial derivative of the x-component of the warp w^t at time t, and let $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ be two corresponding pixels in I_{t-1} and I_t, respectively. The transfer of the warp distortion is then expressed by

$$\frac{\partial w_x^t}{\partial x}(\mathbf{u}_t) = \frac{\partial w_x^{t-1}}{\partial x}(\mathbf{u}_{t-1}). \tag{5.10}$$

This constraint is enforced for the y-component $\partial w_y^t / \partial y$ as well.

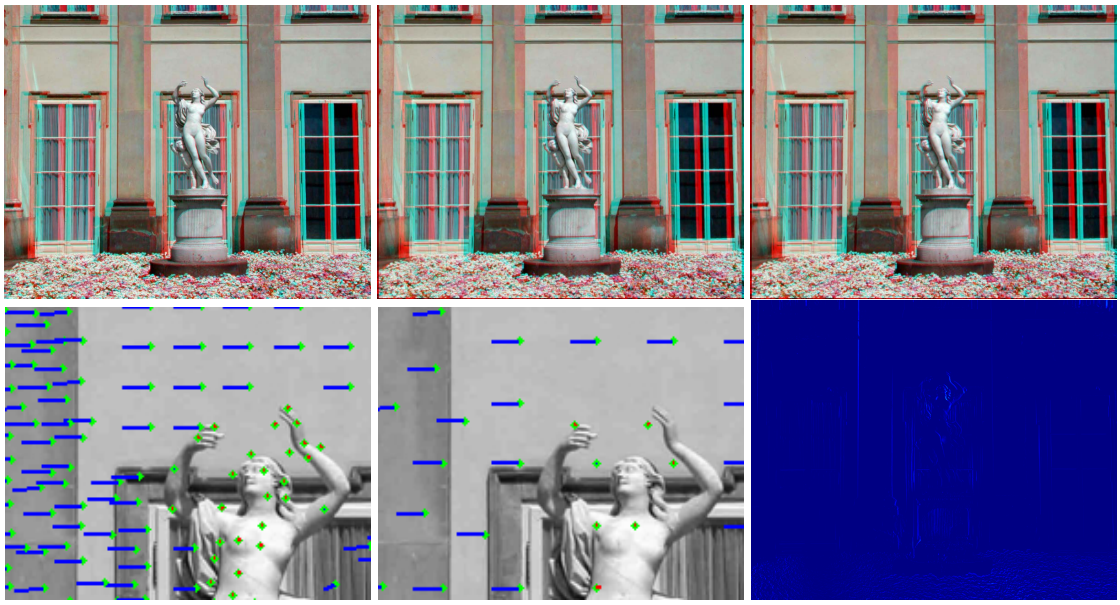


Figure 5.6: *Warping example showing stability over different numbers of stereo correspondences. Upper row, left to right: original stereo input, disparity mapping results (depth range increased) computed with about 2000 features, the same result using only about 200 features. Bottom row: close-ups showing the respective feature density and a difference image of the warped images.*

5.2.6 Saliency Constraints

Besides these novel stereoscopic and temporal warp constraints, we additionally employ the same set of standard constraints of the video retargeting application which minimize the perceivable visual distortion Section 4.2.1. The idea is to enforce a certain rigidity of the warp in salient regions, and to allow for larger image distortions in non-salient regions. Hence, the constraints consist of terms for

- Distortions: $\frac{\partial w_x}{\partial x} = \frac{\partial w_y}{\partial y} = 1$,
- Bending of edges: $\frac{\partial w_x}{\partial y} = \frac{\partial w_y}{\partial x} = 0$,
- Overlaps: $\frac{\partial w_x}{\partial x} \wedge \frac{\partial w_y}{\partial y} > 0$.

During the actual warp computation these constraints are then weighted by the saliency map F_s in Eq. (5.7) to achieve an adaption of the warp to the image-content. Since these basic constraints are identical to the work on video retargeting in the previous chapter, please refer for example to 4 for details.

5.2.7 Implementation

Our implementation of these warping constraints follows the standard procedure in image-based warping 3: The constraints are converted into energy terms so that the computation of the warps (w_l, w_r) can be solved as an iterative nonlinear least-squares problem. In our current implementation we simply sum all of the above energy terms and weight the saliency constraints by multiplication with the saliency map F_s. The warp-induced deformation is illustrated in Fig. 5.4 by overlaying isolines of the original input images. Fig. 5.6 is an example with differing number of stereo correspondences and shows that the results of our stereoscopic warping are quite insensitive to the number of features.

In addition to automatic constraints it would be interesting to include the possibility to manually add high level constraints regarding region positions or global lines (Section 4.2.2). Since at its core our warp is similar to previous warping methods, the inclusion of these techniques is straightforward. However, as we show in our results, the current automatic solution already provides very acceptable results for a variety of stereoscopic 3D footage.

5.3 Results and Applications

As motivated in Section 5.1, the question of how the disparity range should be adapted for different types of stereoscopic footage depends strongly on the particular target application. Our goal was to achieve practical disparity retargeting that can be employed in actual application scenarios. Hence, we present nonlinearly and linearly mapped results for three important application scenarios. We also evaluate the quality of our method quantitatively by a ground truth comparison, and present the results of a user study to validate the perceptual quality of the warping.

The production scenarios we present in this section include nonlinear and linear editing for post-production, automatic disparity correction, and display adaptation. Furthermore we illustrate the versatility of our method with an example for 2D to 3D conversion. We also quickly summarize how our disparity mapping can be used for multi-view (i.e., more than two views) broadcast and auto-stereoscopic displays.

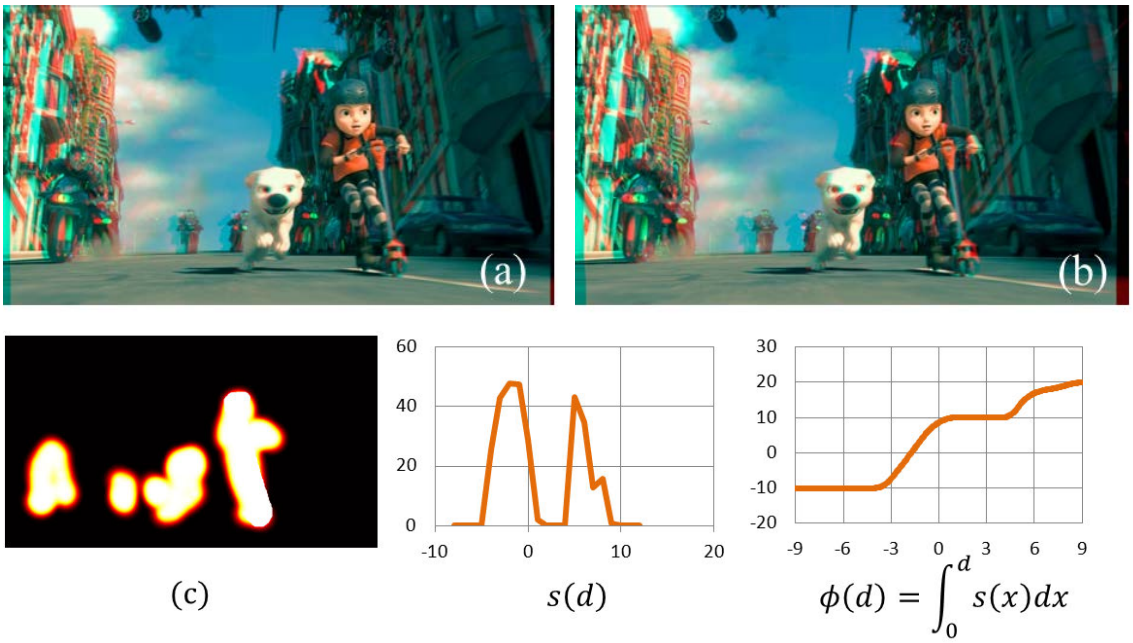


Figure 5.7: *Post-Production. For an input frame (a) and a given importance map (c) the average importance for individual disparities $s(\sigma)$ can be computed. This is automatically converted into a nonlinear depth mapping operator $\phi(\sigma)$ as described in Section 5.1.2. The resulting image in which the key characters are emphasized by stronger depth is shown in (b).*

5.3.1 Post-Production

The first major application area of our disparity mapping operators and warping is post-production of stereoscopic content. We may assume a studio environment where skilled operators apply software tools within an interactive workflow to edit previously captured material. Using the proposed methodology and algorithms, depth composition can be modified and authored by combining different nonlinear and linear disparity operators.

Examples and results for nonlinear and linear disparity editing are shown in Figures 5.1, 5.7, 5.8, and 5.9. In Fig. 5.1 we modify the global scene depth structure with a nonlinear function which emphasizes foreground content and compresses empty space in-between while retaining the maximum disparity of the background. In Fig. 5.7 we exploit a visual saliency map to automatically design an adaptive disparity mapping operator. It enhances salient regions and simultaneously compresses the depth of unimportant regions (see also Eq. (5.3)). In Fig. 5.8 the images were captured with a consumer 3D camera (Fuji Finepix 3D). The resulting disparity range in combination with negative parallax is too large for a comfortable viewing experience on large

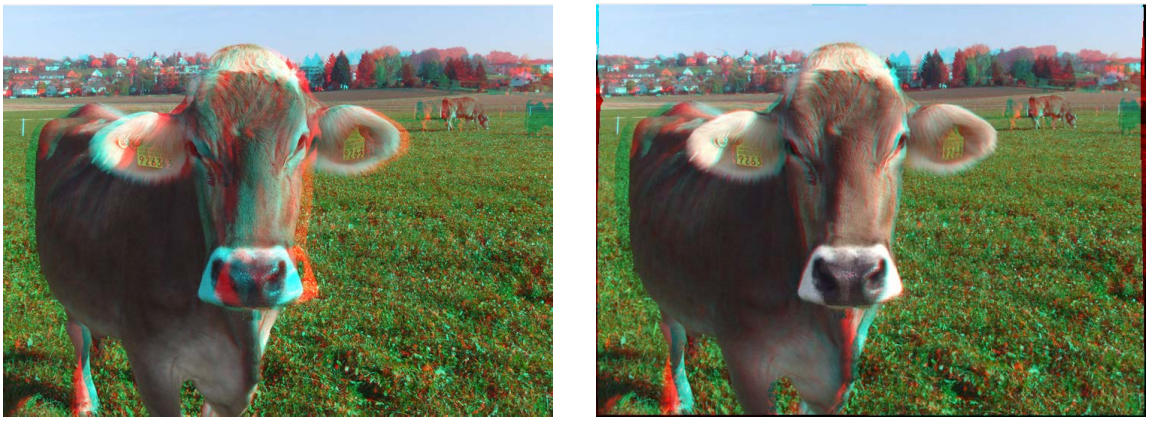


Figure 5.8: *Nonlinear disparity remapping. The disparity range of the original (left) is quite large leading to diplopia on large screens. Our nonlinearly remapped image (right) displays the cow behind the screen and compresses the depth range without apparent flattening of the cow's head.*

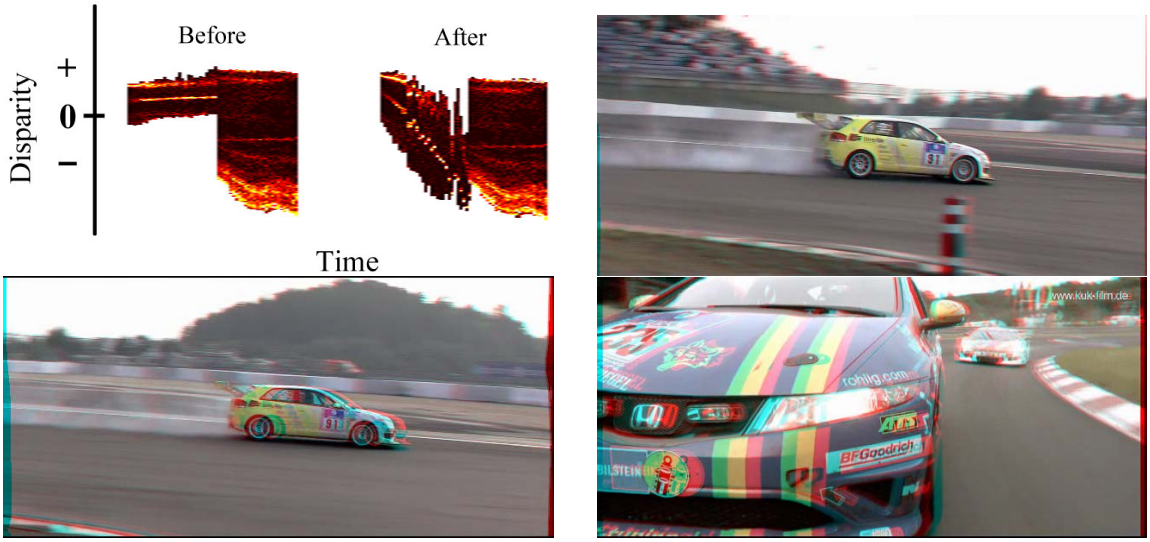


Figure 5.9: *Temporal adaptation for a scene cut. Before adaptation both sequences have considerably different disparity histograms with a clearly visible discontinuity in-between (upper left). Our temporal disparity mapping operator adapts the disparity range of the first sequence (first frame of the sequence in the upper right) to the disparity range of the second sequence (bottom row) and thus achieves a smoother transition.*

screens. We compressed the depth nonlinearly by moving the cow behind the screen surface (positive parallax) and in addition applied a discontinuous nonlinear map retaining the dimensionality of the cow's head without altering the maximum disparity. A further example is shown in Fig. 5.6.

Another important scenario in stereoscopic content production relates to the temporal adaptation of the depth structure in scene transitions and the focus on salient objects therein (Section 5.1.1). Currently, cinematographers are designing depth storyboards in advance and modification of convergence by global image shift is the only tool to smooth over shot transitions. Our methods enable us to compensate for the sharp disparity jumps by slowly adapting the disparity ranges of the previous and/or current scenes. Fig. 5.9 depicts the disparity histograms before and after correction.

5.3.2 Automatic Disparity Correction

Stereography in live action content production is a difficult art. Camera parameters such as baseline and vergence have to be adjusted carefully to ensure a high-quality view experience while keeping the overall action within the stereoscopic comfort zone. Settings are adjusted to match a certain depth range in which the action is expected to take place. In movie productions such decisions are taken by the directors, can be adjusted as appropriate, tested, and shots are repeated if necessary.

This is not possible in live broadcast scenarios or for the amateur home user. Any error will immediately lead to degradation in viewing quality or even result in diplopia. 3D sports broadcast is a popular and timely example. Movements of camera and objects are fast, spontaneous, often unpredictable, and interleaved with rapid scene cuts. This frequently leads to violations of the stereoscopic window or transgressions of admissible disparity ranges. Similar considerations apply to stereoscopic footage captured by amateurs. Simple shift convergence for correction will not help if the overall depth range of the scene is sufficiently large. Instead, careful limitation and compression of the disparity is required. Fig. 5.10 displays examples for automatic disparity correction of such content.

5.3.3 Display Adaptation

A third application area is 3D display adaptation and retargeting. It is motivated by the observation that 3D content optimized for certain target display

Figure 5.10: *Automatic correction of disparity. The original stereo pairs are shown on the left, our result is on the right. The cropped racing car captured with strong negative disparities and a large overall scene depth results in the so-called framing problem. With a simple linear disparity scaling the car is pushed behind the screen without increasing the background disparity. In the bottom example, taken with a consumer 3D camera, the subject was moving towards the camera and finally exceeded the maximum disparity range. In our result, the global disparity range has been adapted so that the background remains at constant depth while the foreground is pushed closer to the screen.*

size and viewing distance (e.g., theatrical) will appear differently on a different medium (e.g., 3DTV). In order to retain a high viewing quality and the artistic intention, disparity adaptation is necessary when reformatting 3D content, e.g., from theatrical to TV or even to a handheld device. Examples of depth editing are illustrated in Fig. 5.13.

5.3.4 2D to 3D Conversion

In order to demonstrate the versatility of our method we illustrate an example for 2D to 3D conversion. Recreating 3D stereo pairs from existing 2D images or video involves an expensive and cumbersome interactive workflow. Re-

Figure 5.11: *This shows 8 views that were synthesized from a stereo image pair. The 3rd and the 6th images are the original input, while the first two and the last two views are obtained by extrapolation and the 4th and 5th view are obtained by interpolation.*

cently, Guttmann et al. [GWC09] presented a novel approach that simplifies this task by solving for dense depth maps from sparse user scribbles to generate stereographic sequences. Using our method the requirement for dense depth can relaxed, so that the sparse scribbles alone are already sufficient. The warp interpolates the pixel disparities and generates a stereographic image pair from a single input image (see Fig. 5.14).

5.3.5 View Synthesis for Multi-View & Auto-Stereoscopic Displays

Stereoscopic 3D is already mainstream, and almost all new display devices for the home support stereoscopic content. However, the necessity to wear glasses is often considered as an obstacle, which hinders broader acceptance of this technology in the home. Multiview autostereoscopic displays enable a glasses free perception of 3D content for several observers simultaneously, and support head motion parallax in a limited range. In order to support multiview autostereoscopic dispays in an already established stereoscopic distribution infrastructure, a synthesis of new views from only two view video is needed. This application briefly describes how disparity aware video warping can be used as a view synthesis method which synthesizes new views directly from stereoscopic two view video and functions completely automatically. We discuss this application as it is an important application for our novel introduced image domain warping method. However, most of the details an evaluation can be found in dedicated papers [SLS12] and [Ste+13].

Multiview Displays

Multi-view auto stereoscopic displays (MAD) enable multiple viewers to enjoy 3D content without the necessity of wearing active or passive glasses. They enable glasses free stereo viewing by emitting several images at the same time. This usually works with parallax-barrier or lenticular lenses. The MAD technology ensures that each viewer in front of the display sees only that stereo pair which is appropriate for his particular viewing position. MADs support also motion parallax viewing in a limited range, i.e. occluded scene parts become visible while other parts are again occluded when a viewer moves his or her head. To achieve these advanced functionalities, MADs require not two but many different views as input. Typical MADs which are on the market today require 8 up to 28 views as input. Because of the different number of input views required by different MADs, no unique display format exists for such displays. Additionally, most 3D content nowadays is produced with two views. Therefore it is essential to have a technology to produce a given number of new views between (interpolation) and around (extrapolation) of two given views.

View Synthesis

For the view synthesis application we compute N-view video from 2-view video where $N > 2$. The first step is to compute two warps w_l and w_r that warp the respective images to a camera position located in the center between the two input cameras. Therefore, we enforce the following disparity constraints:

$$\begin{aligned} w_l(\mathbf{u}_l) - \mathbf{u}_l &= \frac{\mathbf{u}_r - \mathbf{u}_l}{2} \\ \mathbf{u}_r - w_r(\mathbf{u}_r) &= \frac{\mathbf{u}_r - \mathbf{u}_l}{2} \end{aligned} \tag{5.11}$$

For every detected correspondence pair we enforce that the two warps deforms the input images in such a way that they meet halfway towards each other. In other words we computed two warps that capture the deformation information that is required to warp the left and right image towards each other such in the ideal case they would overlap.

Given this general output-view independent warps, we can then for the N-view synthesis interpolate or extrapolate any new warp by linear interpolation $w_{new} = aw + (1 - a)w_0$. w_0 is the uniform (undeformed) warp and w is either the left or the right warp. a is the interpolation value which depends on the number of required output views. We select w_l or w_r depending on

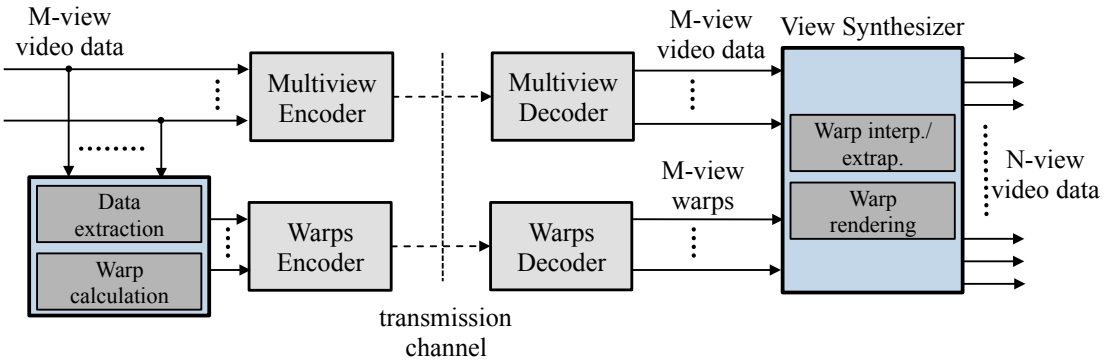


Figure 5.12: *This shows the proposed pipeline of transmission based on warp coding. To reconstruct N-viewes on a remote sides from smaller number M-views we suggest to pre-compute warps on the encoder side and transmit them along with image data. On the remote side only efficient warp interpolation and warp rendering is required. We proposed this in [Ste+13]*

the approximate of the output view to the input views, i.e., we use that input image and that warp that is closer to the desired output view. However, if only one image is used for the synthesis, empty regions can occur on the left or right border of the output image. Hence, for output images located at a position between the input images, texture from a synthesis with the second input image is used to fill the empty border region.

Warp Coding

In practice we found that it is too much of computational challenge and unnecessary for each output display to compute the two warps w_l ansd w_r from scratch to compute its requires output views. Instead we can compute the general warps once and then transmit them along with the video stream to the displays. The displays then are only required to perform the interpolation/ extrapolation step and the warp rendering. Please find a detailed description of this in the following publications [Ste+13] and MPEG standardization proposals [Ste+11].

MPEG Evaluation of Warp Based View Synthesis

N-view synthesis is a highly relevant application for the industry and therefore heavily studied in the MPEG standardization committee. Hence, we used established tests and content for evaluating our view synthesis algorithm. We answered a MPEG call for proposal ([MPE11]) with results generated form

our algorithm. A large subjective study coordinated by MPEG ([MPE12]) showed that multi-view video coding of 2-view or 3-view video in combination with a decoder-side view synthesis based on image domain warping leads to high quality synthesis results without requiring depth map estimation and transmission. Our joint MPEG proposal [Ste+11] was considered as one of the four winning methods for N-view synthesis in this independent study.

5.3.6 Ground Truth Comparison

The purpose of the ground truth experiment was to assess the visual quality of the our warping approach. We utilized a publicly available data-set generated with a multi-camera rig. We picked two views (numbers 8 and 10) from 3 different data-sets and then applied the warp to generate intermediate and extrapolated views (numbers 7, 9, and 11). The extrapolated views represent a doubling of the camera baseline while the interpolated view corresponds to a baseline of 0. We then compared the quality of our result to the known ground truth images using both a perceptually motivated structural similarity metric SSIM [Wan+04] and by computing the RMSE of the difference images. Single pixel shifts can cause a high RMSE error while contributing little to perceived image quality. Hence, perceptual measures like SSIM are generally better suited for such types of comparisons. The values for extrapolated views in Table 5.1 are averaged over the two views 7 and 11. As Table 5.1 and the images in Fig. 5.15 reveal, our method is able to compute interpolated and extrapolated views of a scene that are perceptually indistinguishable from the original, provided the disparity range is not too large. Please note, however, that geometrically consistent view interpolation has not been an explicit goal of this work.

5.3.7 User Study

To further assess the suitability of *warping* for disparity mapping we conducted a user study with 22 subjects and 15 test cases. The goals of the user study were to show specifically that (1) warping indeed results in a perceivable change of a scene's depth structure, and that (2) the quality is not degenerated by visual artifacts.

We performed the study with a line-wise polarized 46 inch full HD display manufactured by Miracube. All 22 participants were tested for their ability to perceive depth on a stereo screen using a random dot stereogram. One subject had to be excluded due to a negative test. The remaining 21 subjects

Figure 5.13: *Display adaptation into both directions. The middle images are the original stereo pairs, while the left images feature a linear reduction of the disparity range to* 50% *and the right images an increase to* 200%*. These results show that our method allows to preserve the initial depth structure relative to the screen geometry to adapt content to actual viewing conditions.*

Figure 5.14: *2D to 3D conversion. From a single 2D input image and providing only a sparse set of disparity cues shown as rough scribbles, our method produces a convincing 3D result.*

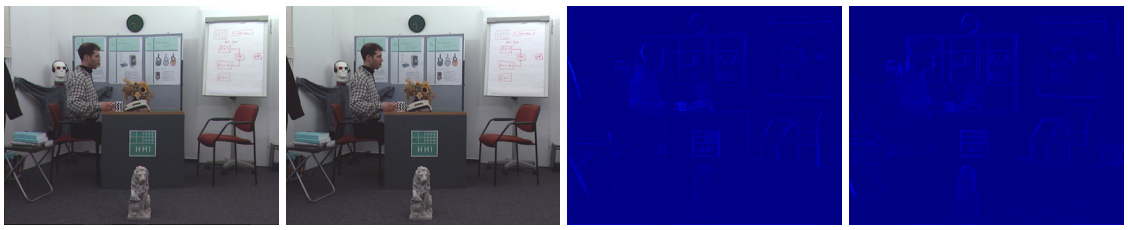


Figure 5.15: *Ground truth comparison showing a known ground truth image (left) versus our warped image (middle left), and absolute difference images of warped and known ground truth for both baseline reduction (middle right) and extension (right).*

took part in a pairwise evaluation [Dav63] of video material composed from

short clips from 3D Hollywood movies, 3D sports and other professional stereo video. Every side-by-side comparison featured the original scene as well as a disparity mapped version of it using our method. For nine test cases we doubled the initial disparities and for six test cases we reduced them by a factor of 0.5. We randomized the order of videos as well as the locations of the original and manipulated videos. For every comparison the participants had to answer the following two questions:

1: Q: Which video features more depth? A: *Left* or *Right*

2: Q: Which video is the original? A: *Left*, *Right*, or *Don't know*

Depth perception

In total we received 311 valid votes regarding the depth impression. Overall, 253 votes (81%) correctly recognized the example with larger disparity range. Kendall's coefficient of agreement [Dav63], which measures the interobserver variability for pairwise comparison tests, is $u = 0.391$ with a p-value < 0.01. Two sequences (street view, train) only reached a correspondence of 66%. Both videos feature a very strong perspective depth cue. One sequence had a recognition rate of only 55%. This sequence contains fast and complex motion cues for scene elements confirming well-known observations in stereo perception, such as the domination of motion parallax cues and the resulting difficulty to focus on binocular depth effects. 17 participants correctly recognized the depth mapping for 11 or more sequences. One subject drastically diverges with only 5 recognitions (9 is the next better result). Removing these outliers (three sequences and one subject) from the evaluation leads a recognition rate of 88%.

Quality

Question 2 was answered in 56% cases as *Don't know*. 25% of the votes correctly identified the original, but 19% were wrong and assumed that the manipulated video was actually the original video. Some originals were correctly recognized by over 80%, but some of the manipulated sequences were also considered originals by over 80%.

The major conclusion we draw from the study is that disparity mapping based on image warping can change the depth structure of a scene in a perceptually believable way without introducing distracting visual artifacts. As we have demonstrated earlier, the suitability of a particular operator (local,

Table 5.1: *SSIM and RMSE values of ground truth comparison with 3 different datasets for view interpolation and extrapolation. SSIM=1 means no difference to the original,*

Dataset	1 interp.	1 extrap.	2 interp.	2 extrap.	3 interp.	3 extrap.
SSIM	.9999	.9998	.9999	.9999	.9999	.9999
RMSE	.0313	.0334	.0235	.0242	.0228	. 0236

global, linear, nonlinear) highly depends on the application, artistic intention, scene content and motion, and other criteria.

5.4 Limitations

A limitation of our method is depicted in Fig. 5.16. For image regions in which the disparity changes rapidly, such as around the pigeons' heads, the sparse features and warp can lead to visible distortions. These artifacts could be addressed by using a higher feature count, denser depth information, or by adding manual constraints to enforce feature preservation. Such methods have been successfully proposed in work on image warping [Krä+09].

However, it is very interesting to observe that, when viewed in 3D, such artifacts are often visually less apparent due to the complex compensation mechanisms of our visual system. These phenomena clearly deserve additional research in the context of stereoscopic warping. From research on media retargeting it is also well known that there are certain limits for warping-based methods on how strongly image content may be deformed before artifacts become visible [Wan+09a]. But since most often the required deformation for disparity adaptation lies in the range of only 1-2% of the overall pixel resolution, these limits are typically not reached. In none of the examples presented in this chapter did we observe visible artifacts, which is further proved by the user study. Finally, our method is limited in the extent to which we can modify the camera baseline for nearby objects, since such operations imply explicit handling of occluded areas to avoid conflicting cues.

5.5 Conclusions

In this chapter we presented a set of *disparity mapping operators* providing a basic formalization of perceptually motivated and production-oriented rules

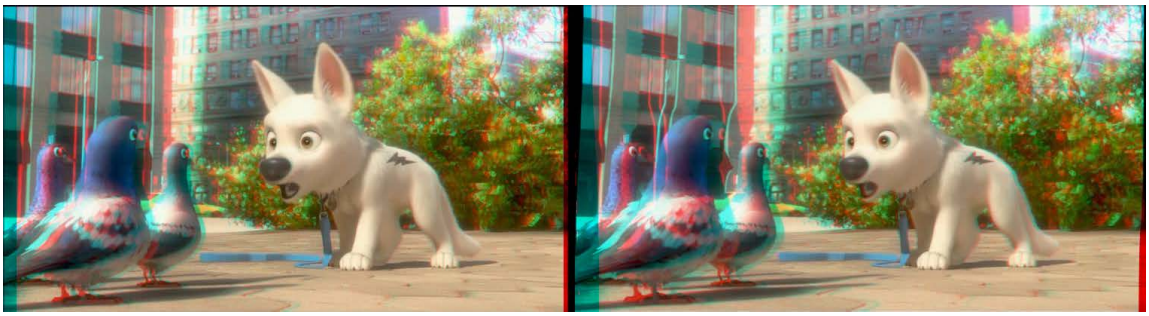


Figure 5.16: *Limitations. For image regions with frequent and strong changes in disparity, the sparse features and the warp can lead to distortions visible around the pigeons' heads. Interestingly, however, such distortions seem to be compensated to some extent by our visual system during stereoscopic viewing.*

and guidelines for nonlinear disparity editing of stereoscopic 3D content. In order to implement these operators we proposed a novel technique based on *stereoscopic warping*, which allows us to deform input video streams in order to meet a desired disparity range. We applied our techniques to three different scenarios, all of which are of very high practical relevance in stereoscopic production and display, and demonstrated that automatic image-based warping could be used as a general alternative for rendering even complex depth manipulations. The quality of stereoscopic warping was evaluated with a ground truth experiment and a user study.

Automatic disparity correction could be implemented in future generation stereo camera systems to support the cinematographer or cameraman in realtime. In addition to professional live broadcast systems, consumer electronic systems could also benefit from such methods, since amateur users generally do not have the experience and background required for proper stereo capture. Moreover, algorithms for disparity adaptation could be implemented as part of future 3D display devices or TVs enabling viewers to control disparity on-the-fly in order to match the display size and the user preferences, much like we control aspect ratio, contrast, or color today. In future research we would like to refine our nonlinear mapping operators to accommodate additional features of stereoscopic perception.

The current user study was limited to assess the suitability of warping for disparity mapping. Future studies will investigate the influence of various nonlinear and local operators on the perceived quality of the results. Furthermore, we want to investigate to what extent conflicting cues, such as inaccurate occlusions, can be compensated for by additional cues like motion parallax.

We also applied the disparity aware image warping to the problem of view

synthesis for autostereoscopic displays. We could show that by warping two input views to a desired number of output views one can achieve very convincing glass free 3D impressions when viewed on a multi-view display. We also proposed to precompute the required warps on the production site and stream it along with the video information to the multi-view enabled receivers to avoid expensive client side computation. This technique is among the favorite for future MPEG standard of multi-view coding.

In this chapter we argued that computation of dense depth maps is often not required. We showed that sparse correspondence detectors together with our warping framework achieve good results compared to DIBR (depth image based rendering) methods. However, sometime dense depth or disparity maps are still valuable. For example the disparity gradient based mapping ϕ_∇ is much more meaningful for a dense map. However, such dense disparity maps are highly difficult to compute especially for video. Therefore, the next chapter will introduce a novel method to compute temporally stable dense maps from sparse input samples.

CHAPTER 6

Efficient Temporal Video Regularization

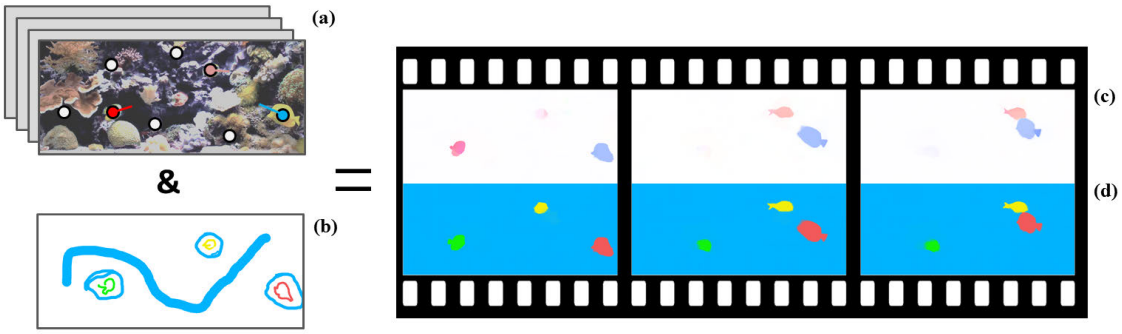


Figure 6.1: *One application of the method described in this chapter is the temporally consistent propagation of scribbles through video volumes. Sparse feature correspondences from an input video (a) are used to compute optical flow (c). Then, color scribbles (b) are spread in space and time to compute the final coherent output (d).*

In this chapter we present an efficient and simple method for introducing temporal consistency to a large class of optimization driven image-based computer graphics problems. We show how the method of domain transform [GO11] can be extended to enable edge preserving filtering to be efficiently computed for video sequences while correctly following (and simultaneously computing) motion vectors in an iterative framework. Our extension

in edge-aware filtering approximates costly global regularization with a fast iterative joint filtering operation. Using this representation, we can achieve tremendous efficiency gains both in terms of memory requirements and running time. This enables us to process entire shots at once, taking advantage of supporting information that exists across far away frames, something that is difficult with existing approaches due to the computational burden of video data. Our method is able to filter along motion paths using an iterative approach that simultaneously uses and estimates per-pixel optical flow vectors. We demonstrate its utility by creating temporally consistent results for a number of applications including optical flow, disparity estimation, colorization, scribble propagation, sparse data up-sampling, and visual saliency computation.

The main difficulty for all these applications is that the spatial-temporal regularization term enforcing smoothness creates dependencies between pixels, often resulting in a large non-convex optimization problem over the entire video sequence. To achieve the required computationally efficient solution that enables *practical* temporal consistency, we trade off accuracy for efficiency, and solve a simpler approximation of the global optimization. In return we can consider more information simultaneously and thereby get a very accurate approximation.

This method extends on the automatic algorithms presented in previous chapters to compute saliency maps, optical flow, edge maps, disparity maps, etc. In the previous chapters we introduced simple frame-wise or trivial window based (couple frames before an after current frame) algorithms. In this chapter we go further into the topic of computing temporally stable dense meta information maps of various kinds which in turn can be used for any previously presented image warping application.

6.1 Method

We address a class of problems that can be solved by minimizing error functionals of the following form,

$$\min_F E(F) = E_{data}(F) + \lambda E_{smooth}(F) \tag{6.1}$$

for unknown solution F, where E_{data} is the application specific error term, and E_{smooth} enforces neighborhood smoothness. The unknown F thereby is a dense map of information over a video sequence. In our discrete video setup F is build up from multiple frames F_t where each of them stores some information $F_{\mathbf{p},t}$ for each pixel $\mathbf{p} = (x,y)^T \in \mathbb{R}^2$ at frame t.

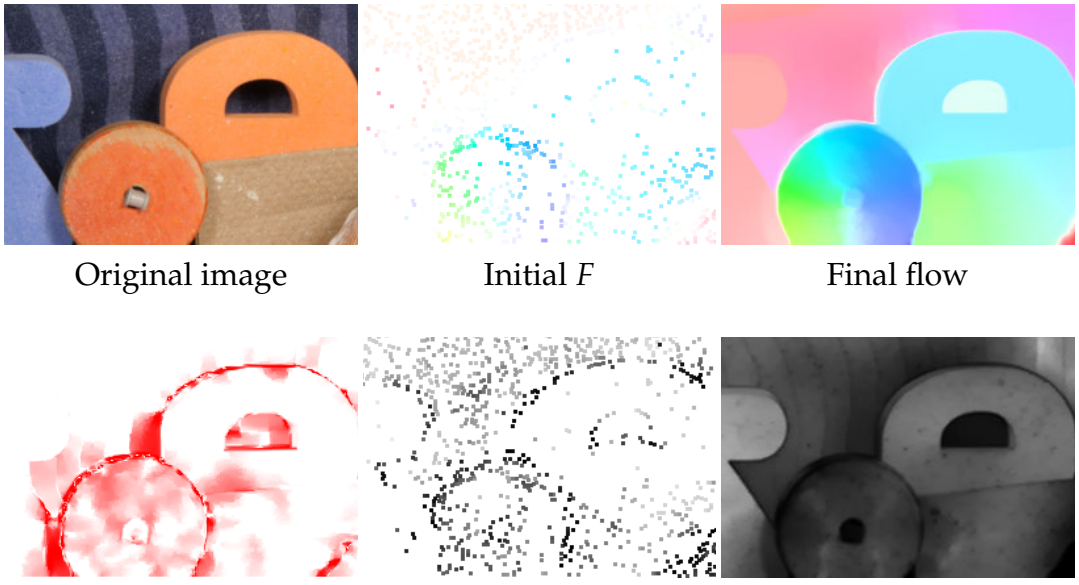


Figure 6.2: *Steps of our method used for solving for optical flow. We start with sparse initial estimates in F provided by feature matching (eroded for clarity). We then use our filtering approach to achieve the final result. The second row shows the occlusion weights ρ, confidence image G and the filtered confidence image.*

The main idea of our method is in order to avoid costly global optimization, we split up the data and smoothness terms and solve them each in series. We do this by first initializing F with application specific initial conditions that minimize E_{data} locally. This initialization should be accurate but does not need to be dense.The regularization term in the energy minimization (E_{smooth}) is then replaced by an efficient edge aware filtering operation on F. In this sense, smoothness is *created* as postulated, rather than solved for with an optimization. Proper formulations of this idea can be tailored to suit a variety of application scenarios which we describe in Section 6.2.

We will begin by explaining how our method works for a single frame, using optical flow as an example application. In this case, our unknowns $F_{\mathbf{p}}$ are set of motion vectors $\mathbf{f} \in F$, expressed as $\mathbf{f}_{\mathbf{p},t} = (u,v)$ corresponding to the motion between two images I_t and I_{t+1} at pixel $\mathbf{p}$. Eq. (6.1) commonly becomes with $\mathbf{p} = (x,y)$ for the optical flow application:

$$E_{data}(u,v) = \sum_{(x,y)} ||I_t(x+u, y+v) - I_{t+1}(x,y)||^2 \tag{6.2}$$

$$E_{smooth}(u,v) = \sum_{(x,y)} (||\nabla u_{xy}||^2 + ||\nabla v_{xy}||^2) \tag{6.3}$$

where ∇ is the gradient magnitude operator, E_{data} encodes the matching cost incurred by F, and E_{smooth} minimizes the total quadratic variation of the flow gradient.

Common algorithms compute the solution to this problem by energy minimization, which can be computationally expensive. Processing video sequences at high resolution while ensuring temporal consistency quickly becomes impractical and hampers application of powerful ideas from image-based graphics. We solve it by replacing global optimization with a local smoothing operation. While this has been studied before [Cri+10; Rhe+11], we find it helpful to quickly go over the mathematical justification to more clearly define the class of applications that our method can be applied to.

By viewing optical flow as a reaction-diffusion problem, we can see that Eq. (6.3) is the Dirichlet's energy, and minimizing it is equivalent to solving the Laplace equation $-\Delta\mathbf{f} = 0$ given E_{data} as a boundary condition. This leads to the following related heat equation

$$\frac{\partial \mathbf{f}}{\partial t} = \alpha \Delta \mathbf{f} \tag{6.4}$$

with the Dirac function initial conditions:

$$\mathbf{f_p} = \begin{cases} F_{\mathbf{p}} & \text{if } \exists \mathbf{p} \in F \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{6.5}$$

In the isotropic case, the Green's function solution to Eq. (6.4) on an infinite interval is simply a Gaussian convolution. As images form an inhomogeneous medium, the arising nonlinear-PDEs can be solved by using anisotropic diffusion [PM90], which in the discrete setting has been shown to be asymptotically equivalent to edge aware filtering [PKT09]. As such, we can see that regularization equations of the form of Eq. (6.3) can be solved with edge-aware filtering, given the right initial conditions (which are determined in the optical flow case, by Eq. (6.2)).

We therefore start by initializing F with sparse feature correspondences computed between frames I_t and I_{t+1} (Eq. (6.5)). In our implementation, we use an out-of-the-box feature matcher from OpenCV that employs SIFT and Lucas-Kanade features. This provides accurate $\mathbf{f}$ vectors in F, but only at locations where reliable estimates can be found. We then compute an edge-aware filtering of F to create the final result (see Fig. 6.2).

In order to realize a temporal edge-aware filtering operation, we extend recent work, called the domain transform [GO11], which we will briefly describe here, but refer the reader to the original work for a complete description. Rather than varying the filter weights based on image content, the signal is

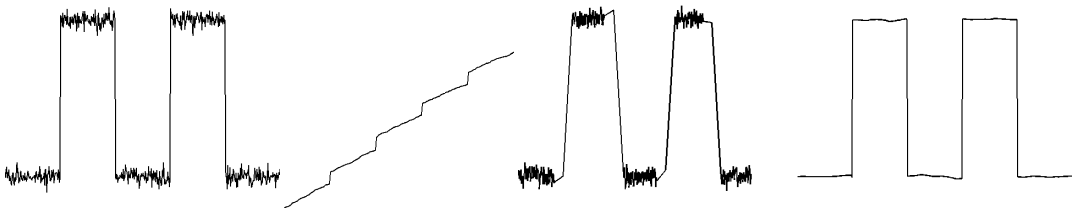


Figure 6.3: *1D domain transform example. (a) Input signal, (b) Transformed x-coordinates, (c) Transformed signal $\hat{C}$, (d) Output after filtering (c) with a Gaussian and re-mapping it to the original domain.*

transformed so that it can be filtered by a fixed width Gaussian (a much more efficient, and importantly *separable* operation). It is then transformed back into the original domain, where strong gradients are maintained. Intuitively, transformed coordinates are computed for each pixel such that two pixels belonging to the same object have nearby coordinates, while pixels that lie on different sides of a strong image gradient are far apart.

Given a regularly sampled 1D signal $\tilde{F}$ and some uniform sampling $S = x_0, x_1 ... x_n$ that defines a curve $C = (x, \tilde{F}(x))$, the domain transform computes a new irregularly sampled curve $\hat{C} = (ct(x), \tilde{F}(x))$, where $ct(x)$ is the transformed coordinate of x, and $ct(x_i) - ct(x_{i+1})$ is the absolute value of the arc-length between points x_i and x_{i+1} on the curve C (or in image space, the geodesic distance on the $RGBXY$ manifold). The coordinates ct can be computed as:

$$ct(u) = \int_0^u (1 + \gamma |\nabla \tilde{F}(x)|) dx \tag{6.6}$$

where γ influences the balance of the spatial versus color distances in the transformed domain. For edge aware smoothing, $\gamma = \frac{\sigma_s}{\sigma_r}$, corresponding to the variance of the filter over the spatial and range domains respectively. Practically speaking, this stretches the abscissa of C, based on its arc-length. $\hat{C}$ can then be filtered with a Gaussian kernel and remapped back to the original domain by inverting Eq. (6.6).

Fig. 6.3 illustrates this applied to a 1D signal.

In the 2D image case, a series of N 1D iterations is performed, alternating between X and Y until convergence. We call the dense filtered solution F'. We perform a joint domain-transform filtering, where the transformed coordinates are determined from image I, and are then used to filter the solution F.

As the Gaussian kernel is not an interpolation kernel and F may be sparse,

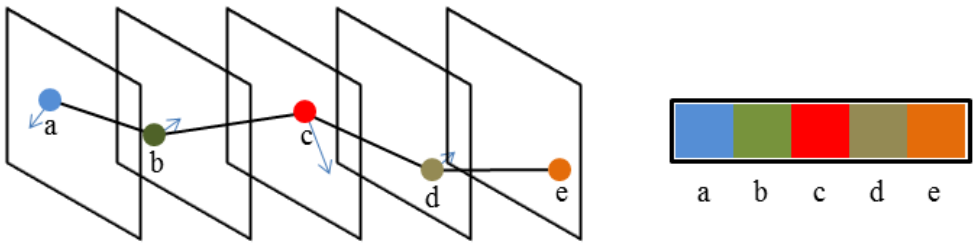


Figure 6.4: *A path (black line connecting pixels) is defined by the motion vectors at each current frame (blue arrows). Filtering in the temporal direction happens along these paths; the resulting 1D vector on the right is iteratively convolved with a box filter in the transformed domain.*

the sum of the kernel weights at each pixel will not necessarily sum to one. A normalization image G is therefore created such that for each pixel i, $G_i = 1$ if there is data at i, and $G_i = 0$ otherwise. This image is filtered with the same filtering operation as F, producing G', which is the sum of the kernel weights per pixel (and inversely proportional to the geodesic distance). The normalized final result is then computed as $F'' = \frac{F'}{G'}$.

We now describe our novel extensions to this method.

6.1.1 Temporal Filtering

We extend the single frame method to video volumes by adding an additional pass of the separable box filter that filters the temporal (T) dimension. Unlike with spatial passes, temporal filtering should follow the motion of points between frames. This prevents incorrectly averaging information across object boundaries and improves results (Fig. 6.6). To enforce temporal continuity, and introduce a temporal smoothness assumption, we desire edge-aware filtering also in the temporal direction. To correctly model motion, our method creates dense estimates of optical flow, regardless of the final application. We call the vector of pixels that correspond to the motion of one scene point over time a *path*. One such path is shown in Fig. 6.4. Paths are computed by following optical flow vectors at each frame (rounding to the nearest pixel), and are then filtered after undergoing a 1D domain transform, similar to a row or column in the spatial filtering passes.

Of course, when computing optical flow, this creates a chicken-and-egg problem; per-pixel motion vectors are required in order to correctly compute the motion vectors! We resolve this using an iterative approach that begins with a rough estimate of the flow as computed by our sparse feature matching

and one spatial X and Y filter pass. This rough estimate provides initial motion vectors for the first temporal pass, which are then updated in each of the N iterations (consisting of one X, Y, T pass each).

We use a sliding box filter for efficiency, as only one addition and subtraction operation is needed for each pixel along the path, regardless of kernel size. However, we must be careful when filtering the temporal direction, as paths do not create a bijective mapping from frame to frame (which is different from rows and columns). This is due to three cases: 1) A path can leave the image boundary, 2) Multiple paths can converge on the same pixel, and 3) A pixel can have no paths in the previous frame that map to it. To compute unbiased results when filtering temporally, it is important that at any given frame, every output pixel belongs to exactly one path. To accomplish this, we begin at the first frame with one path per pixel and maintain a double-ended queue per-pixel that keeps track of the box-filter contents as it moves along that path. At each frame as we move in time and the paths begin to diverge, we find all pixels that no longer belong to a path (due to 1) or 3)). For these pixels, we spawn a new path centered at that pixel, and initialize the box filter by stepping backwards in time by the box-filter radius (in the transformed domain). In case 1), the path is ended and the sliding box filter stopped. In case 2) when multiple paths collide, we randomly keep one while cutting the other off at the previous frame. As the scale of the temporal domain is different, we use different parameters to control γ and the filter radius, which we call σ_{rt} and σ_{st} (all parameters are given in a table at the end of Section 6.2). Not only does enforcing temporal consistency provide more useful results for most applications, it also allows us to fully use all the information available in a video sequence, and provides us denser estimates of E_{data} for filtering. For example, in the optical flow case we can propagate initial feature matching samples from previous or later frames to the current one if the are no (temporal) edges in between.

6.1.2 Confidence

By extending the role of the normalization image G, we introduce a simple way to add confidence values to our data term. As we mentioned before, setting $G(x,y)$ to 1 in the presence of sparse data yields a normalization image that conveys how much known data has reached each pixel. To instead assign a confidence weight β_i to the sparse feature at pixel i, we set the value of $G_i = \beta_i$, and multiply the data image by the same $F_i = G_i F_i$ for all pixels i. We write this per-element multiplication as $G \cdot F$. Again, we filter both G and $G \cdot F$ and compute the final result as $\frac{(G \cdot F)'}{G'}$. In addition to being used for

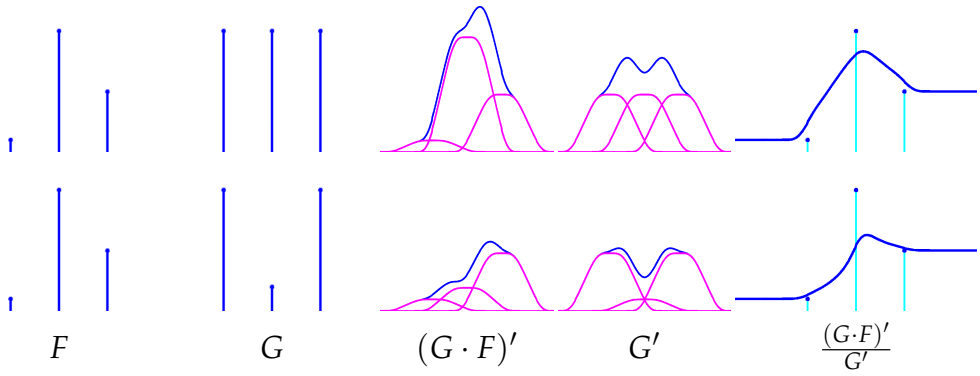


Figure 6.5: *A demonstration of how our confidence term is applied to two 1D examples (one each row). The original sparse signal F is modulated with the confidence G and filtered to produce $(G \cdot F)'$ (blue = signal, pink = contribution of individual points). G' is the filtered confidence image. The final normalized result $\frac{(G \cdot F)'}{G'}$ is shown with the original signal overlaid. In the bottom row, the decreased contribution of the middle point is visible due to its lower confidence value.*

normalization, the confidence term correctly encourages higher confidence points to contribute a greater influence on the final result. This effect is illustrated in Fig. 6.5.

For optical flow and disparity estimation, we use the feature matching vector difference as a confidence weight, increasing the contribution of the sparse features that had better matches. This greatly improves the quality of the results, as shown in Fig. 6.6.

6.1.3 Iterative Occlusion Estimates

We also incorporate additional information into the role of the normalization image G, which helps greatly with occlusion regions. To compute an occlusion likelihood, we estimate both forward and backwards flows ($\mathbf{f}^f$ and $\mathbf{f}^b$) at each frame, and apply a confidence penalty (ρ) for each pixel i based on how well the vectors match. We use a robust penalty function:

$$\rho = (1 - |\mathbf{f}^f + \mathbf{f}^b|)^\theta \tag{6.7}$$

where θ is a parameter that controls the shape of the penalty curve. While computing optical flow, these occlusion estimates ρ change each iteration (ρ^0 to ρ^{N-1}), so we update our occlusion weights based on the flow estimates from the previous iteration. Beginning with sparse data confidence $G^0 = \beta$,

for each iteration n before filtering, we compute $G^n = G^{n-1} \cdot \rho^{n-1}$, and update $F^n = F^{n-1} \cdot \rho^{n-1}$ equivalently. This term has the effect of lowering the confidence in regions where our flow estimates prove unreliable. This in turn causes higher confidence spatial temporal neighbors to exert a greater influence on these occlusion regions. The occlusion information is used to weigh the confidence of not only the flow, but also the additional data terms in the corresponding applications we introduce later.

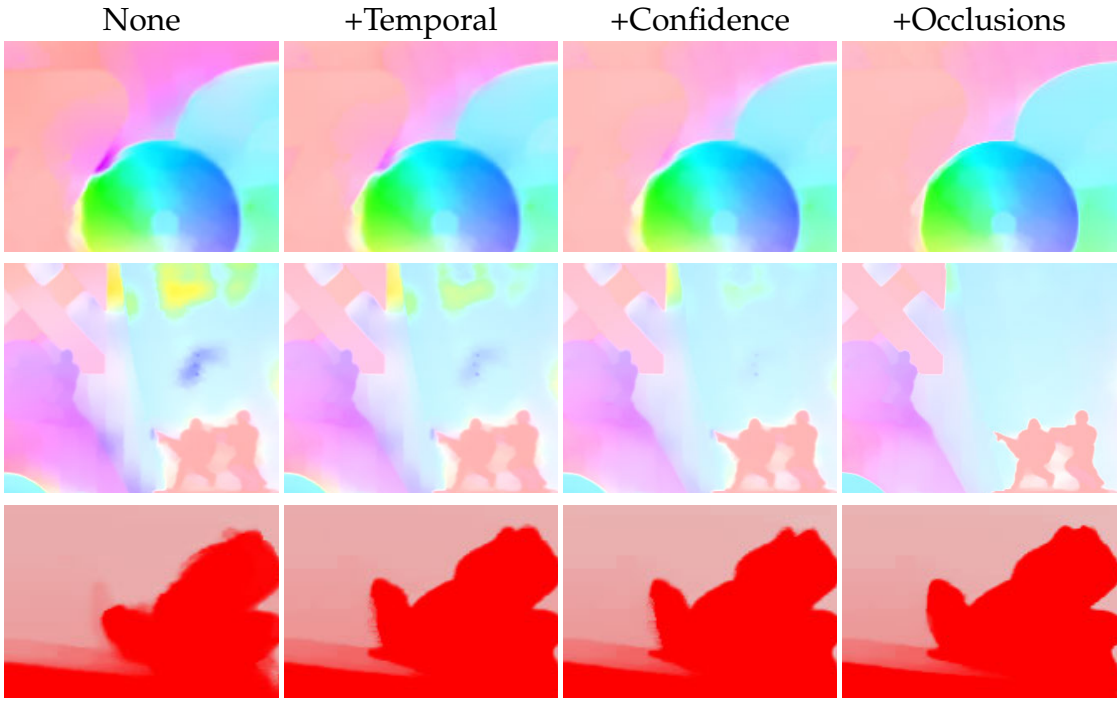
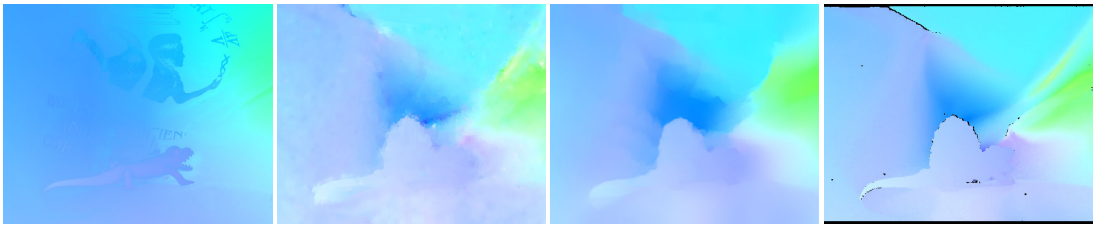


Figure 6.6: *Table showing the effect of each processing step introduced. We incrementally add temporal filtering, confidence values, and flow occlusion estimates to optical flow, and scribble propagation.*

6.1.4 Evaluation

We show the improvements gained by each of the three steps in Fig. 6.6 on several datasets, for the applications of optical flow (top two) and scribble propagation (bottom). In the first column, we show the results using the naive solution of a temporal edge aware filter (filtering the temporal dimension straight through the video volume). We then incrementally add our motion path adhering temporal filtering, confidence values, and occlusion estimates. The edge regions in particular are greatly improved, as are incorrect values that arose due to noise in our initial matching (such as on the flat background surface in the central row). In the scribble propagation example, we can see

improvements in the outline of a frog. We also compare our results to other approaches for joint-data upsampling in Fig. 6.7, showing that our method outperforms a bilateral filter [CPD07], and a global solution computed using a locally-linear assumption [LLW06]. The bilateral filter allows flow to spread incorrectly through similarly colored regions, and the global solution exhibits noise due to the ℓ_2 penalization of incorrect feature matches.



Bilateral Filter	Global Solve	Our Method	Ground Truth
RMSE 0.42	RMSE 0.21	RMSE 0.12	

Figure 6.7: *A comparison of different methods for propagating sparse correspondences to compute optical flow. The results shown here were generated using each authors' publicly available code, and choosing parameters that yielded the lowest RMSE between ground truth and estimated flow vectors (in pixels).*

Quantitatively validating temporal optical flow is difficult, as public ground-truth datasets for long, real world sequences are not readily available. The widely used Middlebury ranking provides short eight-frame sequences with one frame of ground truth for testing. However, as noted by previous temporally aware methods [ZBW11] due to capture methodology, these sequences exhibit large, temporally discontinuous motions that violate a smoothness assumption. Such prior works therefore avoided making comparisons to this data with temporal methods. Similarly, we found that our method visually performed significantly worse under these conditions than with real-world footage. However, for comparison, we include the latest Middlebury rankings; at the time of submission, our method had an average rank of 56.1 in terms of average endpoint error.

Additionally, we compare our results on real world video datasets to a number of existing state of the art methods; one that works on single frames [ZBW11], and two temporal approaches that use sliding windows, one variational [Vol+11] and one cost-volume filtering method [Hos+11] (currently ranked 10th, 5th, and 15th in Middlebury evaluation respectively at the time of submission). We visualize the comparison both with color coded motion vectors, and then most significantly, by directly viewing the result of using optical flow in a typical application, in this case frame-rate upsampling.

An additional advantage of our method is that by using a descriptor-based

feature matcher, we are able to correctly detect small objects that exhibit large motion, as long as they contain suitable features for matching. This is something that is traditionally very difficult to model, as most variational methods linearize the image, meaning that the result is only valid locally. Long range motion is detected by computing over a scale-space pyramid, which can cause small objects to disappear.

However, we note that our motion-path adhering temporal filtering method is separate from the feature-matching component, and any method for filling E_{data} could be used, even existing optical flow approaches. Therefore, our method can be used as a post-filter for any optical flow method. And thereby also allowing those methods to take advantage of information (flow vectors) in many frames.

Additionally, please note that for applications other than optical flow we are not required to use our flow computation. Instead, we can use any optical flow method to construct motion paths and use those to filter other applications informations with our method.

6.2 Applications

We now apply our method, enforcing temporal smoothness on a number of different applications.

6.2.1 Optical Flow

As explained before, we compute the optical flow with our method regardless of the application such that we can filter along the motion paths. Besides that, often the optical flow is then not further used. However, the optical flow itself of course is already an important application. We did multiple experiments to evaluate the quality of our optical flow.

In each of the following cases, we first compute optical flow and then compute the application-specific result, using the above described filtering approach.

6.2.2 Disparity Estimation

Disparity estimation involves computing dense correspondences between a rectified pair of stereoscopic images. For this application, solution $F_{xy} = \sigma_{xy}$ contains scalar values that describe the disparity between a stereo image pair I^l and I^r at pixel (x, y). Similar to optical flow, this can be expressed as:

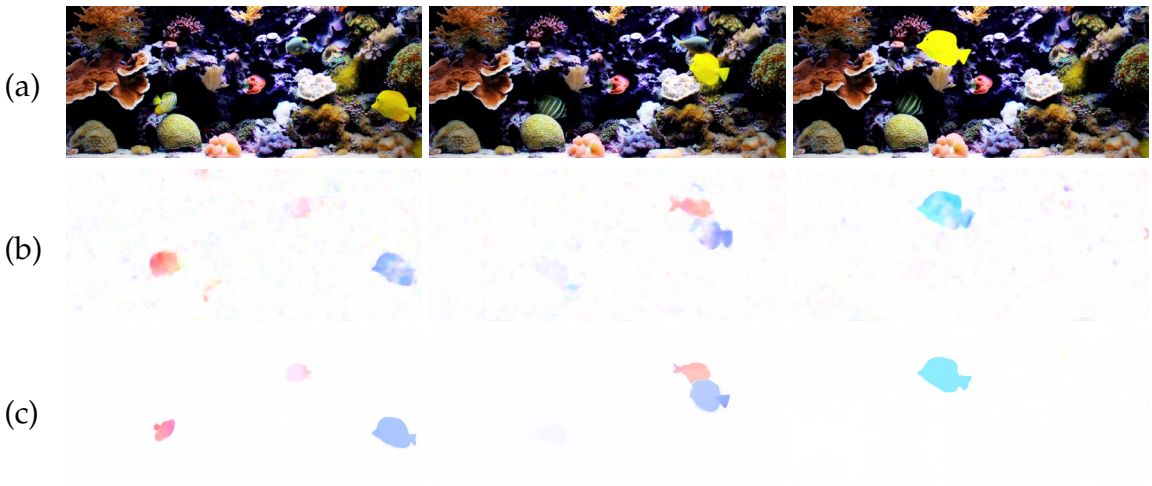


Figure 6.8: *Given an high resolution (1280x544) input video (a), optical flow is compared to a high ranking (10th in the Middlebury evaluation at time of submission) per-frame method (b) [ZBW11]. Row (c) shows individual frames of our result.*

$$E_{data}(\sigma) = \sum_{(x,y)} ||I^r(x+\sigma_{xy},y) - I^l(x,y)||^2$$

$$E_{smooth}(\sigma) = \sum_{(x,y)} (||\nabla\sigma_{xy}||^2).$$

We again sparsely compute an initial F using feature matching between I^l and I^r, and perform our filtering. We evaluate the quality of our results by comparison to high quality disparity maps provided with publicly available MPEG test sets [Wil+10]. One advantage of our approach is that by design our disparities are well aligned to image edges. This is an important feature in a number of applications, such as virtual view synthesis and object insertion.

This dense and temporally smoothed disparity maps can now also be used as input to our stereoscopic image warping and disparity mapping. For example the disparity gradient based operator can be more easily implemented for dense disparity maps. Also, the temporal smoothed disparity maps makes the warp computation temporally more stable and allows to reduce the temporal constraint in the warp computation.

6.2.3 Depth Upsampling

Depth sensors often provide information that has lower spatial resolution,

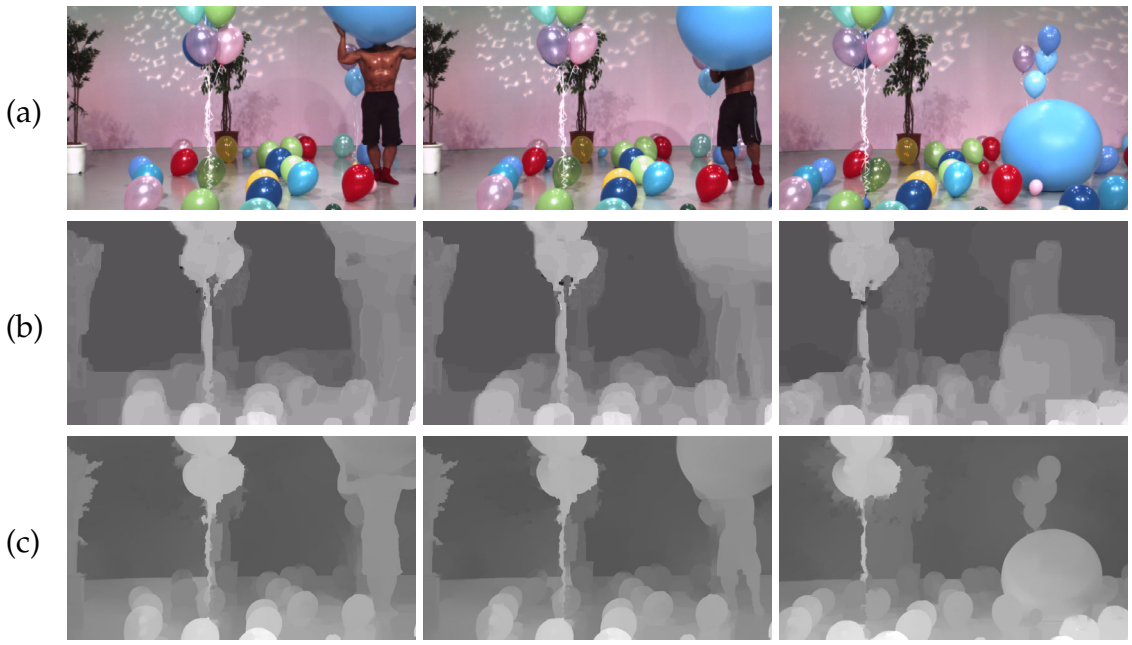


Figure 6.9: *The high resolution stereoscopic input video provided by the MPEG group (a) is used compute a dense disparity map. (b) shows the result of the state of the art algorithms [Wil+10]. Our result is shown in row (c).*

missing data due to parallax between sensors, and is temporally noisy. We address all of these issues by enforcing edge-aware spatial and temporal smoothness on the depth data. We tested our method by initializing F with the depth data from a Microsoft Kinect sensor, and filtering this using the video data to compute our transformed coordinates. The initial data we used were captured from multiple kinect sensor within the project of [Kus+11].

The depth maps are shown before and after upsampling in Fig. 6.11. We can see that our method fills in unknown areas adhering to image edges, and produces temporally consistent results for the entire sequences.

6.2.4 Colorization and Scribble Propagation

Another application of our method is the temporal and spatial propagation of sparse user input. These pen strokes can be used to colorize videos, represent high-level labeling, or to provide scene composition cues that are intuitive to humans, such as depth ordering [Wan+11a]. In prior work, E_{data} enforces the given scribble map, and E_{smooth} assumes a locally-linear model [LLW04],

Figure 6.10: *Here we shows an example of using the depth map only to insert a new object (z-test). Our depth map is better aligned to image edges.*

written for pixels i and scribbles s as:

$$E_{smooth}(s) = \sum_i ||s_i - \sum_{p \in \mathbb{N}(s_i)} w_{pi} s_p||$$

where w_{pi} are the locally linear weights. We initialize $F_i = s_i$, and apply our temporal smoothness assumption to propagate scribbles at key-frames throughout the video. For colorization, we convert the RGB/grayscale image I into YCbCr space and replace the CbCr color channels with those specified by the propagated user scribbles. After replacing the we convert back to the then finally colored RGB image.

The required number of scribbles depends on the amount of motion in the scene, as scribbles are only valid while the scene has a similar composition. We found that in practice, we were able to get convincing results by creating on average one scribble key-frame for every 20 output frames, which is on par with sample results from other global optimization approaches.

6.2.5 Saliency

Determining visual saliency is a very important component of many image-based graphics operations, especially also for the applications presented in

Figure 6.11: *Row (a) shows the input sequence and the used scribbles. Row (b) shows the result of the scribble propagation with our method. These are frames of a long sequence. Scribble input was only given at frame 0 and frame 15. The middle column show the temporal result at frame 11.*

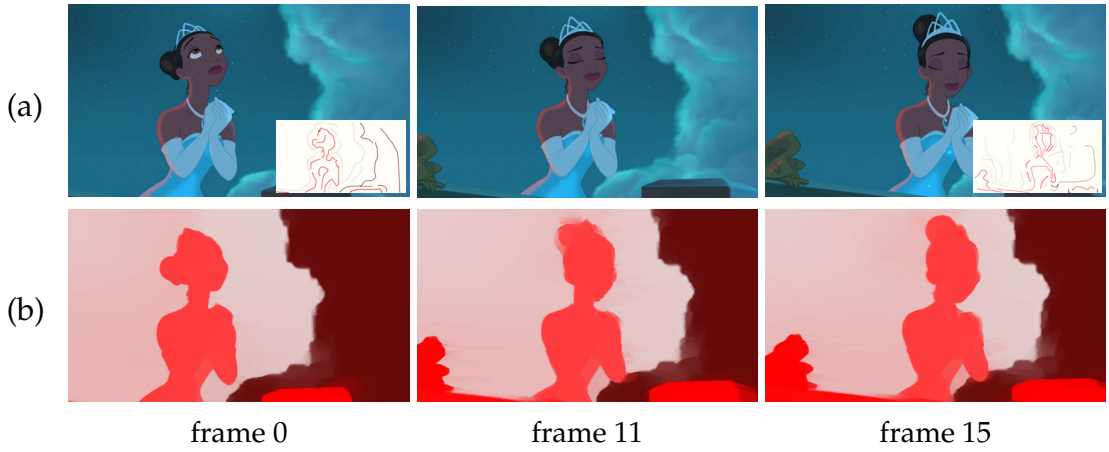
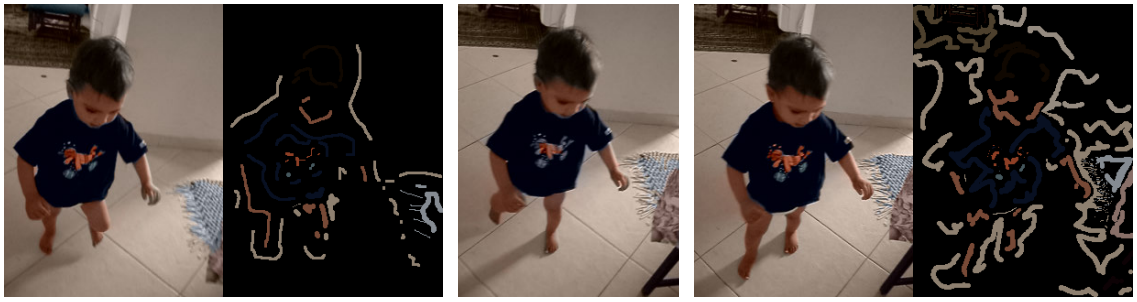


Figure 6.12: *Row (a) shows the input sequence and the used scribbles. Row (b) shows the result of the scribble propagation with our method. These are frames of a long sequence. Scribble input was only given at frame 0 and frame 15. The middle column show the temporal result at frame 11.*

Chapter 4 and Chapter 5. We used efficient frame by frame exist that estimate importance by analyzing the frequency spectrum [GMZ08].However those methods are highly nonlinear response. In particular small changes from one frame to the next can introduce big changes in the computed salience map. Previously, to avoid such highly temporal varying input for the retargeting and disparity mapping project we did use window based averaging. However, now with mehtod described within this chapter we can significantly improve the results by considering image edges and motion path wehn filtering salience maps.

First, we use our default method to compute per-frame saliency l, initializing with $F(x,y) = l(x,y)$, and then introduce our temporal smoothness, producing a cleaned, stable output. We validate our saliency by using it for the



frame 5 scribble 5 frame 13 frame 20 scribble 20

Figure 6.13: *Colorization: We used the data and scribbles of the previous work [Lew+04]. Please note that there is no other scribble input between frames 5 and 20.*

application of video retargeting, where the aspect ratio of a video is modified while preserving the appearance of visually significant objects, see Chapter 4 In this case, the temporal stability of the saliency map is very important, as noise in the map can cause visually distracting wobbling in the final result. With our improved temporally stable and image edge aligned salience we could even completely remove the temporal warp constraint (Eq. (4.9)) and still create temporally stable retargeting results. This especially interesting because it allows to compute per-frame warps without inter-dependencies of each other and therefore allow for better parallelization.

6.2.6 Parameters

We use the following parameter values for all datasets:

Application	σ_s	σ_r	σ_{st}	σ_{rt}	N	θ
Flow	2000	.4	5	.1	4	5
Disparity Estimation	2000	.3	5	.1	4	5
Scribble Propagation	1000	.3	5	.1	4	5
Depth upsampling	1000	.1	5	.1	4	5
Saliency	20	1.5	15	.2	4	5

6.3 Performance

An important benefit of our method is that it is computationally simple. In addition, the processing steps are mainly independent and the memory accesses mainly local, so it scales well to multi-core or hardware implementations. We

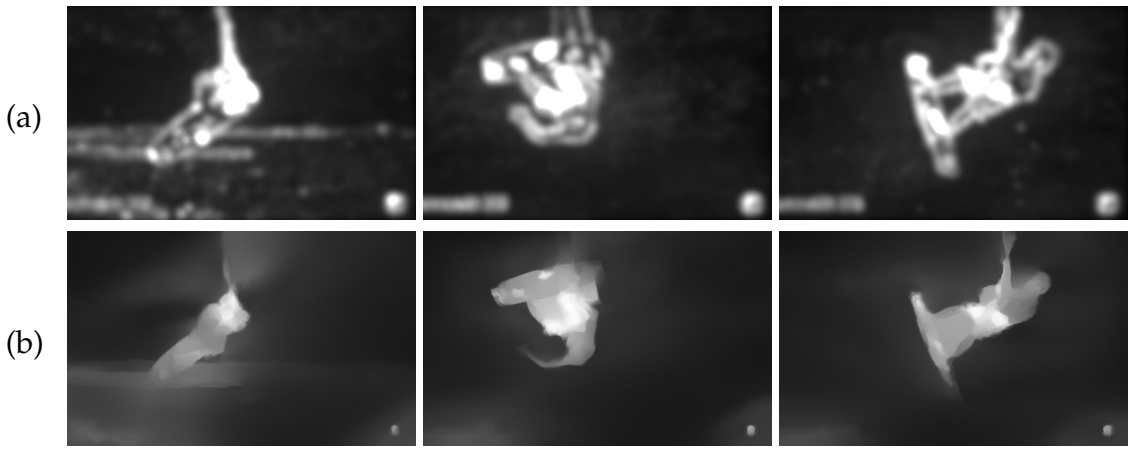


Figure 6.14: *Row (a) shows the input per frame not object aligned saliency of a wakeboard sequency. Row (b) shows the improvement we get by applying our filtering. Per-frame outliers are reduced and the saliency map is better object aligned. Using such a importance map for video warping (Chapter 4 and Chapter 5) can greatly reduce manual interaction and tuning.*

provide all timing information for a HD 720p sequence on a desktop computer (Core i7 920 2.67GHz (4 cores) CPU and 12GB of memory), ignoring file IO time. In the following we will step through our algorithm and describe quit detailed the complexity of each step where "flops" values are estimated by looking at our C implementation. The actual compiler generated number may differ.

First, we load the video sequence and compute spatial transformed coordinates, which stay fixed for the sequence. This takes 3.6ms per frame requiring about 12 flops per pixel,

Spatial filtering is implemented as a sliding box filter requiring approximately (depending on the transformed coordinates) three floating point operations per pixel per data channel. For optical flow, six data channels per pixel (forward and backwards flow (u, v) and normalization channels G) are processed, this takes 35ms per frame. Temporal filtering also is performed with a sliding box filter, but in this case motion paths must be followed. These paths change every iteration, requiring the coordinate transform along a path to be recomputed on-the-fly which requires 16 flops. To filter a path, the sliding box filter stores a double-ended queue of entries, which avoids us from having to re-follow flow vectors as the box translates. We implemented this as a ring buffer that allows for quick push_front and pop_back operations, this occurs once for every pixel in video cube. Additional special cases have to be handled when paths begin and end, due to the cases mentioned in Section 6.1.

When a new paths is created, a sliding box filter is initialized around the current pixel, reconstructing the path backwards in time. This adds a small number of operations, but happens only for a few pixels in the video video (around 2.3%). In that case we need additional half the box filter size number of operations of the general case. While rows and columns can be trivially parallelized, filtering paths requires additional synchronization. We used an atomic check-and-set to ensure that when multiple paths converge, only one thread continues and the other ends. Temporal filtering requires a larger memory footprint to store path queues, and more random access patterns in memory. As such it is slower than the spatial passes, taking 96ms per frame.

Finally, after every XYT iteration, we update the confidence maps. This requires Eq. (6.7) to be evaluated per pixel, which takes 20.7ms per frame.

In total, our proposed spatio-temporal filtering requires approximately 151.6ms per frame per iteration for HD video. For $N = 4$ iterations, and a shot consisting of 100 frames of 720p material, we can perform all filtering operations on six data channels (computing forward and backward flow) in 65.2 seconds; 0.652 seconds per frame.

Our method additionally requires computing sparse (forward and backward) feature correspondences. We do this for all frames in parallel, using an out-of-the-box OpenCV solution that required 145.43ms on average per frame.

We compare our method to publicly available timing information from the existing state-of-the-art optical flow methods that we validate against. Times reported are for the task of computing eight frames of optical flow on a Middlebury sequence (640x480 resolution).

Method	Time per output frame	Total for 8 frames
[Rhe+11]	55 seconds	7.3 minutes
[ZBW11]	620 seconds	1.4 hours
[Vol+11]	40 minutes	5.4 hours
Our method	625 ms	5 seconds

Our method does not use sliding windows, and takes only 5 seconds on a standard CPU for the 8 frames, 3.2 seconds for initial feature matching and 1.8 second for our proposed spatial-temporal filtering. Please see the video for a comparison of result quality.

Our method is also efficient in terms of memory usage, requiring order O(N) memory, where N is the number of pixels in the video volume. This allows long sequences to be computed simultaneously. Processing 400 frames of 640x480 video, or 140 frames of 1280x720 video requires 8.83 GB memory with

our naive implementation. Our current implementation is mainly memory limited, although with current hardware we had no problems running our method on high resolution long sequences that ended up using 100GB of memory. And given the easy parallelization of our method we could easily handle such large examples by utilizing 16 cores simultaneously.

However, as most computational steps are trivially parallelizable, longer videos could easily be supported by swapping images in and out of memory.

6.4 Limitations

Of course, our approach is not without limitations. Primarily, we have decided to trade accuracy for computational efficiency, in the process greatly simplifying the problem that we are solving. One consequence of this is our dependence on having sufficient initial conditions; it is possible that objects can remain undetected when there are not enough image features to match between frames. To alleviate this, we tuned the parameters of our feature matcher to find *as many* features as possible, even when this creates a number of bad matches, as our filtering approach suffers more from the lack of data than from the presence of outliers (incorrect matches), which are largely filtered out by spatial and temporal neighbors. This works well if the confidence measurement of the feature detector returns reasonable values.

Our method also shares similar limitations to most image-based computer graphics approaches in that it can fail when important object boundaries are not well represented by image edges. However, in applications with very sparse input such as scribble propagation, greedy mistakes can cause data to incorrectly bleed across boundaries, something that would be avoided in a true global solution. As a result, when compared to prior scribble propagation work, our method requires scribbles to be more localized at object boundaries; this is a trade-off for our efficiency gains. This effect can be seen in Fig. 6.13, where we perform colorization using scribbles from a prior global approach [LLW04]), and show a comparison between our result and theirs. One possible solution could be to user interaction where the more important object edges are highlighted and the confusing texture edges are suppressed. As our method can provide interactive rates for single key-frames and fast results for video sequences, this kind of interaction could greatly improve the results with minimal overhead.

Despite these limitations, it is our hope that this approach will open the door for the practical application of many existing and future image-based computer graphics techniques to video data.

6.5 Conclusion

In summary, we have presented a simple and efficient approach for approximating global smoothness over video sequences using temporal edge-aware filtering. Our method is robust and achieves stable results over a variety of datasets using a fixed set of parameters per application.

By introducing a temporal smoothness assumption, we have shown that it is possible to obtain good results for difficult image-based computer graphics problems such as optical flow and disparity estimation by just using a feature matcher and filter in place of costly global minimization.

Our method has not been fully optimized for speed, and further performance gains could be made by exploiting GPU parallelism. Recent work describes significant performance gains computing summed area tables on the GPU, which is a large component of our computation [Neh+11].

We also showed how this spatial-temporal filtering allows to improve the quality of before mentioned video warping algorithms. By filtering saliency maps F_s we can reduce warp computation complexity by removing the temporal warp constraint and therefore decoupling frame computations. Replacing the sparse feature detector for stereoscopic video warping with temporally stable dense disparity maps allows us to design even better disparity mapping operators as well as reducing artifacts resulting from temporally unstable input.

CHAPTER 7

Conclusion

In this thesis, we introduced a novel framework for efficient video processing, and demonstrated its relevance and practicality by implementing and analyzing various applications for video adaption, modification, and synthesis. We first observed that many practical applications can be expressed as an image deformation process, which we called an image warp. Recognizing this similarity we started first by introducing a solid, extensible and novel IDW framework. We showed that by carefully crafting the discretization of the warp function we could lay the foundation for a simple but versatile application-specific constraint development. We then discussed how given the discretization and the constraints we can construct an energy optimization problem and efficiently find the desired image deformation using a custom iterative solver targeted specifically for the properties of image warping.

Knowing how to deform an image is only the first step and followed by actually performing the deformation on an image, which we called warp rendering. We carefully analyzed the mathematical process of warp rendering. We showed that the traditional way of backward mapping (often implemented using bilinear sampling on the GPU) is non-optimal from a signal processing point of view. It can easily create aliasing artifacts. Based on this analysis, we consequently introduced a novel warp rendering algorithm based on the EWA framework, which was originally formulated only for 3D rendering. We reviewed its mathematical derivation and approximations and showed how to adapt the framework to 2D warp rendering. We could then

show that EWA based rendering is not only simple and efficient to implement on a GPU, but considerably reduces aliasing artifacts for IDW applications.

After establishing our generalized IDW framework, we discussed various video processing applications and showed how these applications can be efficiently solved using our framework. We first analyzed the problem of video retargeting as the problem of changing the aspect ratio of a given video and thereby hiding necessary deformations imperceptibly in less important image regions, while keeping the aspect ratio of more important regions uniform. We extended previous work in important ways. We introduced a global scaling factor such that multiple salient objects in an image could be scaled the same way. We also showed how to formulate warping constraints that faithfully handle image edges and scene cuts. Besides those automatically computed constraints we introduced user controllable tools to artistically direct the desired outcome of a retargeting process. For example, we introduced tools to enforce straight lines as well as to control the position of objects in the image. We verified the quality of the video retargeting application by an online user study and could show that our approach is superior to linear scaling. More importantly, users rated our video retargeting algorithm superior to other computationally more expensive methods.

In a next step, we extended the retargeting process to stereoscopic imagery and 3D displays. In addition to aspect ratio changes, retargeting for 3D entails also modifying the depth composition of stereoscopic content. We started the discussion with a basic formalization of perceptually motivated and production-oriented rules for nonlinear disparity editing of stereoscopic 3D content. Based on these rules and guidelines, we introduced the concept of non-linear, local disparity mapping operators. We continued by proposing that such disparity-modifying operators can be applied using image domain warping only. We presented how to design 3D-IDW constraints for stereoscopic content such that the deformed image pair will fulfill the postulated disparity mapping operators. Using 3D-IDW, we were able to change the disparity of 3D content much more reliably than previous methods, without the need for computationally expensive and error-prone steps such as dense depth map reconstruction or occlusion in-painting. We carried out a user study which showed that our method can indeed alter the depth impression of content without introducing disturbing artifacts.

Being able to change a depth impression does not only allow for display adaptation, but we showed that this is a much-desired editing tool in practice. Beside other applications, we presented how to optimize scene cuts for stereoscopic content, or how to compress disparity to avoid card-boarding. Therefore, our 3D-IDW framework provided a highly useful, semi-automatic

tool that allows to non-linearly edit the perceived depth impression of content after it was captured. We also introduced how to compute additional interpolated and extrapolated synthetic views for 2-view video, which is vital for auto-stereoscopic displays. Moreover, we also showed a first conceptual pipeline for 2D to 3D conversion based on IDW.

While the IDW framework allowed to solve a wide variety of video-processing problems in a temporally-stable manner, it only was applicable to video applications that required image warping. We therefore introduced a more general energy optimization framework that achieves practical temporal consistency for a broader range of video analysis and processing applications. We analyzed recent advances in fast edge-aware image filtering and extended and combined those methods for video. By considering that many global data regularization tasks can be approximated with filtering, we could achieve significant speed-ups for many relevant video applications. In particular, we showed that our temporally consistent filter framework could be applied to a wide range of different applications such as optical flow, dense disparity map estimation, depth upsampling, scribble-based video colorization as well as saliency analysis. Our framework was not only able to achieve plausible temporal consistency, it also improved the quality of the results much more efficiently than other approaches for video.

7.1 Future Work

The methods and results presented in this thesis were just the start of many interesting follow-up projects. Some of these follow-up projects were carried out in parallel to this thesis, after our initial papers were published, some of which were carried out in collaboration with us. In the following, we will discuss some of these projects and possible future work of our research.

General Image Domain Warping and Warp Rendering We introduced a novel image warping framework and EWA based rendering technique that works well in practice. Nevertheless, it would be interesting to further improve the optimization scheme used for finding the final warp function. It would be interesting to introduce the mathematical tools that allow estimating convergence behavior of the different application constraints. Although we already achieved interactive frame rates with our iterative solver it is likely that a more advanced solver (e.g., by utilizing preconditioners) could improve result quality or performance even further. Also, the proposed EWA based rendering method could be analyzed and optimized in more

detail. Particularly, using Gaussian filters, although convenient, is not a very good approximation of an ideal low-pass filter. Replacing the Gaussians with a sharper filter could help to reduce some over-blurring when rendering. It could be interesting to analyze how such a more complicated pixel splatting function would be deformed by an non-linear warp and if such an advanced splatting operation could be implemented on recent GPUs. The thesis of Greisen [Gre13] carefully analyzed our EWA rendering and proposed some modifications to the anti-aliasing filter to avoid excessive blurring. Furthermore, [Gre13] adapted some of our IDW work for dedicated hardware implementations and showed that our framework can efficiently be implemented in FPGA and ASIC prototypes.

Video Retargeting Video retargeting is still a very actively researched area. With our initial publication [Krä+09] we achieved interactive results for streaming video for the first time. Although such interactive results are important, it would be interesting to see if more complex algorithms could achieve better temporally stable results. In fact, some follow-up work on improving temporal consistency [Wan+10; Gru+10]. Our implementation achieved high performance by utilizing the GPU. While mobile GPU also become more powerful it would be interesting to find algorithms that are better tailored to mobile handhelds. This may not only include simplifying the computation ([Gre+12b], [PWS12]) but also consider the perceptual circumstances of very small displays. In addition to an aspect ratio change, video retargeting often entails a significant resolution change. Although we did optimize the warp rendering to reduce aliasing in such cases, one could further investigate how extreme resolution changes (e.g., 4k to SD) affect other parts of the retargeting pipeline. Furthermore, the quality of our retargeting framework depends heavily on the quality of the video analysis tools such as saliency, edge detection etc. Better algorithms for video and scene understanding could further improve the quality of our retargeting pipeline.

IDW based Disparity Mapping In our discussion of stereoscopic perception we established the general framework of disparity mapping operators. We see some future research opportunity in analyzing and proposing new specialized incarnations of such mapping operators. Similar to research of tone mapping one can imagine to have a large set of well studied, maybe application or artistically defined disparity mapping operators introduced in the future. Furthermore, we found that IDW is very often a good approximation of complex DIBR methods and, moreover, even introduces less

noticeable artifacts than re-projection and in-painting algorithms used in DIBR. However, this is only valid for small changes in disparity (such as stereoscopic content produced for broadcasting or cinema), but is not necessarily true for significant changes in disparity. In such situations it would be interesting to see if a combination of traditional DIBR approaches and our IDW-based methods could improve results. Similarly a discontinuous warp (combined with in-painting) like in [Wan+11b] could be used to increase the range for disparity mapping. Another interesting research topic is to extend our stereoscopic editing to the field of stereoscopic compositing, i.e., combining multiple stereoscopic video sources into one video. One particularly interesting subtopic would be to also match vanishing points in such a case (similar to what was proposed for 2D images by [CAA10]). We have already but together some initial comparisons of different 3D-IDW based compositing methods in [Sch+12], but a more in-depth analysis would still be interesting future work. Similar to video retargeting a dedicated hardware implementation on ASICs or FPGAs would be an interesting step forwards, as it would allow to integrate disparity mapping into consumer electronics products (TV). While our user study showed that local, non-linear disparity changes did not introduce any disturbing artifacts, it would be interesting to perform further user studies to investigate the impact of different disparity mapping operators on the depth perception and to devise a set of optimal mapping operators for different scenarios. Additionally, it would be interesting to have a user study that independently investigates disparity mapping operators from the way they are applied (in our case 3D-IDW). Moreover, a stereoscopic rendering engine or ray-tracer that allows to directly embed different non-linear disparity mapping operators would be a very interesting avenue for future research. Such a rendering engine would avoid any artifacts that can occur with image-based warping and could be used for the aforementioned user studies which only focuses on the disparity mapping operators. But certainly such a rendering extension would also be an invaluable tool for animation-movie production or 3D gaming.

Practical Temporal Consistency for Video Given that our algorithm is suitable to solve a large number of image processing problems, one can focus future research on any of those applications. For optical flow, we got very promising results, and especially our object edge accuracy was very high. For initial feature tracking, however, we used a standard out of the box method. It would be interesting to test if better initialization would result in better motion vector end-point or angular error. Additionally, our method does not support sub-pixel accuracy, and it would therefore be a nice future contribution to see if our temporal filtering could be extended to

fully utilize and compute sub-pixel accuracy for flow vectors . In contrast to entertainment content, computer vision applications often have more than two views of video available. In such a situation one could investigate how to improve disparity and depth computation, but also scribble and color propagation by utilizing all the data in all available views. One can speculate that this adds an extra dimension, which could be handled efficiently similar to our temporal path based filtering.

Furthermore, high-definition content with high spatial and high temporal resolution requires novel approaches and algorithms to cope with the explosion in complexity, and previous state-of-the-art methods often become infeasible or even fail for such high-resolution content. As we showed in our work, improving the computational efficiency per pixel while at the same time using more information makes such large problems more tractable. We think that reconsidering traditional algorithms such as Sobel Filters for high-definition content using similar approaches to our work, provides a wide and auspicious field for future research.

APPENDIX A

Notation

This appendix covers the used notation.

A.1 General Mathematical Notation

$\mathbb{N}$	Set natural numbers.
$\mathbb{R}$	Set of real numbers.
$\mathbb{R}^n$	Set of real n-vectors.
$\mathbb{R}^{m \times n}$	Set of real $m \times n$ matrices with m rows and n columns.
$\mathbf{a} \in \mathbb{R}^n$	Bold lowercase letters are denoting column of any dimension.
$\mathbf{a}_x$	The x component of $\mathbf{a}$
$\mathbf{A} \in \mathbb{R}^{m \times n}$	Bold uppercase letters are denoting matrices of any dimension.
$\mathbf{A}^T, \mathbf{a}^T$	Denoting a transposed matrix or vector
$f(\ldots)$	A function
$f^{-1}(\ldots)$	The inverse function
$g \otimes f$	Convolution of function g with f

A.2 Image Warping

I_I, I_O	Input and Output Image
W, H	Width and height of the input image
$\mathbf{u}$	Position in the input image domain
$\mathbf{u}_k$	Position of the k'th input pixel
$\mathbf{p}$	Position in the output image domain
$\mathbf{p}_i$	Position of the i'th output pixel
$w(\mathbf{u})$	Warp function from input to output position ($\mathbb{R}^2 \rightarrow \mathbb{R}^2$)
$\mathbf{p}_k = w(\mathbf{u}_k)$	Output(warped) position of the k'th input pixel
$\mathbf{w}^n$	Grid vertex position of n'th grid corner of the discretized warp function
$\mathbf{w}_x^n$	x-component of grid vertex n
$\mathbf{J}_k$	Jacobian of warp function at position $\mathbf{u}_k$
$\mathbf{d}^{\mathbf{x}}(\mathbf{u}) \approx \frac{\partial w(\mathbf{u})}{\partial x}$	Finite differences approximation of the warp in x-direction at position $\mathbf{u}$
$d_y^x(\mathbf{u})$	y-component of the aprox. partial derivative in x.
$E(w)$	Energy function which enforces constraints on the image warp
c_{s1}	Constraint (energy term) enforcing a specific property $s1$
c_{s1}^k	Constraint at specific location $\mathbf{u}_k$
$\mathbf{w}^{new}$	During iterative solving a new updated grid vertex positions
$\mathbf{w}^{old}$	Grid vertex position in previous iteration
$\mathcal{N}(\mathbf{w}^{old})$	Previous positions of all neighbor vertices
$q_{\mathbf{w}}(\mathbf{w}^{new})$	Weight for updating $\mathbf{w}^{new}$ with neighbor $\mathbf{w}$

A.3 Warp Rendering

I_I^k ..... Input intensity at input pixel k
$g_I()$ ..... The interpolation function for input pixels
$f_s(\mathbf{u}) = I_i \otimes g_I$ ..... The continuous pixel intensity function of the input image
I_O^i ..... Output intensity at output pixel i
$h()$ ..... Low-pass anti-aliasing filter for the output domain
$G_{\mathbf{V}}(\mathbf{d})$ ..... 2D-Gaussian function with covariance matrix $\mathbf{V}$

A.4 Video Retargeting

w_t ..... Warp function at time t
s_w, s_h ..... Desired vertical and horizontal scaling
s_f ..... the uniform scaling factor for important image regions
F_s ..... The saliency map of the input Image
t_c ..... Binary value to indicate if frame t is a scene cut
cog ..... The center of gravity of a 2D polygon.
loc ..... The desired center location of a polygon
α, b ..... Unknowns that define a line (angle, offset)

A.5 Disparity Mapping

(I_l, I_r) ..... Pair of left and right input image
$\sigma(\mathbf{u})$ ..... Disparity at location $\mathbf{u}$
$\Omega(\ldots)$ ..... Disparity mapping function
$\mathcal{F}$ ..... Set of matched features
w_l, w_r ..... Warp of the left and right image

A.6 Temporal Video Regularization

F	Feature map sequence (motion vectors, saliency, etc.)
F_t	Feature map at time t
$f_{\mathbf{p},t}$	Individual feature (motion vector, saliency value) at location $\mathbf{p}$
u, v	Horizontal and vertical component of optical flow
G	Confidence map
F', G'	Filtered feature map and confidence map

Bibliography

[ABD10] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. "Fast High-Dimensional Filtering Using the Permutohedral Lattice". In: *Comput. Graph. Forum* 29.2 (2010), pp. 753–762.

[Ake+04] Kurt Akeley, Simon J. Watt, Ahna Reza Girshick, and Martin S. Banks. "A stereo display prototype with multiple focal distances". In: *ACM Trans. Graph.* 23.3 (2004), pp. 804–813.

[AR07] Amit Agrawal and Ramesh Raskar. "Gradient Domain Manipulation Techniques in Vision and Graphics". In: *ICCV Courses*. 2007.

[AS07] Shai Avidan and Ariel Shamir. "Seam carving for content-aware image resizing". In: *ACM Trans. Graph.* 26.3 (2007), p. 10. ISSN: 0730-0301. DOI: http://doi.acm.org/10.1145/1276377.1276390.

[Bak+11] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. "A Database and Evaluation Methodology for Optical Flow". In: *International Journal of Computer Vision* 92.1 (2011), pp. 1–31.

[BGL04] Martin S. Banks, Sergei Gepshtein, and Michael S. Landy. "Why Is Spatial Stereoresolution So Low?" In: *Journal of Neuroscience* 24 (2004), pp. 2077–2089.

[Bha+10] Pravin Bhat, C. Lawrence Zitnick, Michael F. Cohen, and Brian Curless. "GradientShop: A gradient-domain optimization framework for image and video filtering". In: *ACM Trans. Graph.* 29.2 (2010).

[BHM00] William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A multigrid tutorial: second edition*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2000. ISBN: 0-89871-462-1.

[BJ80] P. Burt and B. Juelsz. "A disparity gradient limit for binocular fusion". In: *Science* 208.4444 (May 1980), pp. 615–617.

[Ble+09] Michael Bleyer, Margrit Gelautz, Carsten Rother, and Christoph Rhemann. "A stereo approach that handles the matting problem via image warping". In: *CVPR*. 2009, pp. 501–508.

[BM04] Simon Baker and Iain Matthews. "Lucas-Kanade 20 Years On: A Unifying Framework". In: *IJCV* 56.3 (2004), pp. 221–255.

[Buc07] Ian Buck. "GPU computing with NVIDIA CUDA". In: *SIGGRAPH '07 Course Notes*. 2007.

[CAA09] Robert Carroll, Maneesh Agrawala, and Aseem Agarwala. "Optimizing content-preserving projections for wide-angle images". In: *ACM Trans. Graph.* 28.3 (2009).

[CAA10] Robert Carroll, Aseem Agarwala, and Maneesh Agrawala. "Image warps for artistic perspective manipulation". In: *ACM Transactions on Graphics* 29.4 (2010).

[Che+03] Li-Qun Chen, Xing Xie, Xin Fan, Wei-Ying Ma, HongJiang Zhang, and He-Qin Zhou. "A visual attention model for adapting images on small displays". In: *Multimedia Syst.* 9.4 (2003), pp. 353–364.

[CPD07] Jiawen Chen, Sylvain Paris, and Frédo Durand. "Real-time edge-aware image processing with the bilateral grid". In: *ACM Trans. Graph.* 26.3 (2007), p. 103.

[Cri+07] A. Criminisi, A. Blake, C. Rother, J. Shotton, and P. H. Torr. "Efficient Dense Stereo with Occlusions for New View-Synthesis by Four-State Dynamic Programming". In: *Int. J. Comput. Vision* 71.1 (2007), pp. 89–110. ISSN: 0920-5691. DOI: `http://dx.doi.org/10.1007/s11263-006-8525-1`.

[Cri+10] Antonio Criminisi, Toby Sharp, Carsten Rother, and Patrick Pérez. "Geodesic image and video editing". In: *ACM Trans. Graph.* 29.5 (2010), p. 134.

[Dav63] H. A. David. *The Method of Paired Comparisons*. Charles Griffin & Company, 1963.

[DDN08] Thomas Deselaers, Philippe Dreuw, and Hermann Ney. "Pan, Zoom, Scan – Time-coherent, Trained Automatic Video Cropping". In: *CVPR*. 2008.

[Dol+10] Jennifer Dolson, Jongmin Baek, Christian Plagemann, and Sebastian Thrun. "Upsampling range data in dynamic environments". In: *CVPR*. 2010, pp. 1141–1148.

[DT05] James Diebel and Sebastian Thrun. "An Application of Markov Random Fields to Range Sensing". In: *NIPS*. 2005.

[ES07] Todd A. Ell and Stephen J. Sangwine. "Hypercomplex Fourier Transforms of Color Images". In: *IEEE Transactions on Image Processing* 16.1 (2007), pp. 22–35.

[EU13] EU. *MUSCADE. FP7 EU Project*. Available online (Accessed 22 Aug 2014). 2010-2013. URL: `http://www.muscade.eu`.

[Far+11] Miquel Farre, Oliver Wang, Manuel Lang, Nikolce Stefanoski, Alexander Hornung, and Aljoscha Smolic. "Automatic content creation for multiview autostereoscopic displays using image domain warping". In: *ICME*. 2011, pp. 1–6.

[Fou10] the Foundry. *Ocular, Nuke*. http://www.thefoundry.co.uk/. Jan. 2010.

[FSK03] Ingo Feldmann, Oliver Schreer, and Peter Kauff. "Nonlinear Depth Scaling For Immersive Video Applications". In: WIAMIS, 2003.

[G+08] Li Guan, Jean-Sebastien Franco, Marc Pollefeys, et al. "3D object reconstruction with heterogeneous sensor data". In: *International Symposium on 3D Data Processing, Visualization and Transmission*. 2008. URL: `http://hal.archives-ouvertes.fr/inria-00349099/`.

[GH86] Ned Greene and Paul S. Heckbert. "Creating raster Omnimax images from multiple perspective views using the elliptical weighted average filter". In: *IEEE Comput. Graph. Appl.* 6.6 (1986), pp. 21–27.

[GMZ08] Chenlei Guo, Qi Ma, and Liming Zhang. "Spatio-temporal Saliency detection using phase spectrum of quaternion fourier transform". In: *CVPR*. IEEE Computer Society, 2008.

[GO11] Eduardo S. L. Gastal and Manuel M. Oliveira. "Domain transform for edge-aware image and video processing". In: *ACM Trans. Graph.* 30.4 (2011), p. 69.

[Gor+96] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. "The Lumigraph". In: *SIGGRAPH*. 1996, pp. 43–54.

[Gre+12a] Pierre Greisen, Manuel Lang, Simon Heinzle, and Aljoscha Smolic. "Algorithm and VLSI Architecture for Real-Time 1080p60 Video Retargeting". In: *Eurographics / ACM SIGGRAPH Symposium on High Performance Graphics* (June 2012), pp. 57–66.

[Gre+12b] Pierre Greisen, Manuel Lang, Simon Heinzle, and Aljoscha Smolic. "Algorithm and VLSI Architecture for Real-Time 1080p60 Video Retargeting". In: *High Performance Graphics*. 2012, pp. 57–66.

[Gre13] Pierre Greisen. "Hardware Architectures for Real-Time Video Processing and View Synthesis". Series in Microelectronics Volume 221. PhD thesis. ETH Zurich, 2013.

[Gru+10] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan A. Essa. "Discontinuous seam-carving for video retargeting". In: *CVPR*. 2010, pp. 569–576.

[GSC06] Ran Gal, Olga Sorkine, and Daniel Cohen-Or. "Feature-aware texturing". In: *Proceedings of Eurographics Symposium on Rendering*. 2006, pp. 297–303.

[GW02] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 2002.

[GWC09] Moshe Guttmann, Lior Wolf, and Daniel Cohen-Or. "Semi-automatic stereo extraction from video footage". In: *Computer Vision, 2009 IEEE 12th International Conference on*. Oct. 2009, pp. 136–142. DOI: 10.1109/ICCV.2009.5459158.

[Hac86] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer Verlag, 1986.

[Hen+07] Anton van den Hengel, Anthony R. Dick, Thorsten Thormählen, Ben Ward, and Philip H. S. Torr. "VideoTrace: rapid interactive scene modelling from video". In: *ACM Trans. Graph.* 26.3 (2007), p. 86.

[HN11] Tracy Hammond and Andrew Nealen, eds. *Sketch Based Interfaces and Modeling, Vancouver, BC, Canada, 5-7 August 2011. Proceedings*. Eurographics Association, 2011. ISBN: 978-1-4503-0906-6.

[Hof+08] David M. Hoffman, Ahna R. Girshick, Kurt Akeley, and Martin S. Banks. "Vergence-accommodation conflicts hinder visual performance and cause visual fatigue". In: *Journal of Vision* 8.3 (Mar. 2008), pp. 1–30.

[HOK11] Matthias Höffken, Daniel Oberhoff, and Marina Kolesnik. "Temporal Prediction and Spatial Regularization in Differential Optical Flow". In: *ACIVS*. 2011, pp. 576–585.

[Hos+11] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, and Margrit Gelautz. "Temporally Consistent Disparity and Optical Flow via Efficient Spatio-temporal Filtering". In: *PSIVT (1)*. Ed. by Yo-Sung Ho. Vol. 7087. Lecture Notes in Computer Science. Springer, 2011, pp. 165–177. ISBN: 978-3-642-25366-9.

[HR02] Ian P. Howard and Brian J. Rogers. *Seeing in Depth*. Oxford University Press, New York, USA, 2002.

[HS81a] Berthold K. P. Horn and Brian G. Schunck. "Determining Optical Flow". In: *Artificial Intelligence* 17.1-3 (1981), pp. 185–203.

[HS81b] Berthold K. P. Horn and Brian G. Schunck. "Determining Optical Flow". In: *Artif. Intell.* 17.1-3 (1981), pp. 185–203.

[HST10] Kaiming He, Jian Sun, and Xiaoou Tang. "Guided Image Filtering". In: *ECCV (1)*. 2010, pp. 1–14.

[IKN98] Laurent Itti, Christof Koch, and Ernst Niebur. "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis". In: *IEEE PAMI* 20.11 (1998), pp. 1254–1259. ISSN: 0162-8828. DOI: http://doi.ieeecomputersociety.org/10.1109/34.730558.

[Kim+08] Man-Bae Kim, Seno Lee, Changyeol Choi, Gi-Mun Um, Nam-Ho Hur, and Jin-Woong Kim. "Depth Scaling of Multiview Images for Automultiscopic 3D Monitors". In: *3DTV08*. 2008.

[Kno+07] H. Knoche, M. Papaleo, M. A. Sasse, and A. Vanelli-Coralli. "The Kindest Cut: Enhancing the User Experience of Mobile TV through Adequate Zooming". In: *ACM Multimedia*. 2007, pp. 87–96.

[Kra+08] Vladislav Kraevoy, Alla Sheffer, Ariel Shamir, and Daniel Cohen-Or. "Non-homogeneous resizing of complex models". In: *ACM Trans. Graph.* 27.5 (2008), p. 111.

[Krä+09] Philipp Krähenbühl, Manuel Lang, Alexander Hornung, and Markus H. Gross. "A system for retargeting of streaming video". In: *ACM Transactions on Graphics, Siggraph Asia* 28.5 (2009). DOI: 10.1145/1618452.1618472.

[Kus+11] C. Kuster, T. Popa, C. Zach, C. Gotsman, and M. Gross. "FreeCam: A Hybrid Camera System for Interactive Free-Viewpoint Video". In: *Proceedings of Vision, Modeling, and Visualization (VMV)*. 2011.

[Lam+09] Marc Lambooij, Wijnand IJsselsteijn, Marten Fortuin, and Ingrid Heynderickx. "Visual Discomfort and Visual Fatigue of Stereoscopic Displays: A Review". In: *Journal of Imaging Science and Technology* 53.3, 030201 (2009), p. 030201. DOI: 10.2352/J.ImagingSci.Technol.2009.53.3.030201. URL: http://link.aip.org/link/?IST/53/030201/1.

[Lan+10] Manuel Lang, Alexander Hornung, Oliver Wang, Steven Poulakos, Aljoscha Smolic, and Markus Gross. "Nonlinear disparity mapping for stereoscopic 3D". In: *ACM Transactions on Graphics, Siggraph* 29.4 (2010).

[Lan+12] Manuel Lang, Oliver Wang, Tunç Ozan Aydin, Aljoscha Smolic, and Markus H. Gross. "Practical temporal consistency for image-based graphics applications". In: *ACM Transactions on Graphics, Siggraph* 31.4 (2012), p. 34.

[Lew+04] Adrian Lew, Patrizio Neff, Deborah Sulsky, and Michael Ortiz. "Optimal BV estimates for a discontinuous Galerkin method in linear elasticity". In: *Applied Mathematics Research Express* 2004.3 (2004), pp. 73–106.

[LG06] Feng Liu and Michael Gleicher. "Video retargeting: automating pan and scan". In: *ACM Multimedia*. 2006, pp. 241–250.

[LH74] Charles L Lawson and Richard J Hanson. *Solving least squares problems*. Vol. 161. SIAM, 1974.

[LH96] Marc Levoy and Pat Hanrahan. "Light Field Rendering". In: *SIGGRAPH*. 1996, pp. 31–42.

[Liu+09] Feng Liu, Michael Gleicher, Hailin Jin, and Aseem Agarwala. "Content-preserving warps for 3D video stabilization". In: *ACM Transactions on Graphics* 28.3 (2009).

[LLW04] Anat Levin, Dani Lischinski, and Yair Weiss. "Colorization using optimization". In: *ACM Transactions on Graphics* 23.3 (2004), pp. 689–694.

[LLW06] Anat Levin, Dani Lischinski, and Yair Weiss. "A Closed Form Solution to Natural Image Matting". In: *CVPR (1)*. IEEE Computer Society, 2006, pp. 61–68. ISBN: 0-7695-2597-0.

[Low04] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[Mah+09] Dhruv Mahajan, Fu-Chung Huang, Wojciech Matusik, Ravi Ramamoorthi, and Peter N. Belhumeur. "Moving gradients: a path-based method for plausible image interpolation". In: *ACM Trans. Graph.* 28.3 (2009).

[Men09] Bernard Mendiburu. *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*. Focal Press, 2009.

[MP04] Wojciech Matusik and Hanspeter Pfister. "3D TV: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes". In: *ACM Trans. Graph.* 23.3 (2004), pp. 814–824.

[MPE11] MPEG-N12036. *Call for Proposals on 3D Video Coding Technology*. SO/IEC, March 2011.

[MPE12] MPEG-N12347. *Report of Subjective Test Results from the Call for Proposals on 3D Video Coding*. ISO/IEC, 2012.

[Neh+11] Diego Nehab, André Maximo, Rodolfo S. Lima, and Hugues Hoppe. "GPU-efficient recursive filtering and summed-area tables". In: *ACM Trans. Graph.* 30.6 (2011), p. 176.

[Neu09] Robert Neuman. Personal Communication with Robert Neuman, Chief Stereographer, Disney Animation Studios. 2009.

[PBP00] Yael Pritch, Moshe Ben-Ezra, and Shmuel Peleg. "Automatic Disparity Control in Stereo Panoramas (OmniStereo)". In: *OMNIVIS*. 2000.

[PD06] Sylvain Paris and Frédo Durand. "A Fast Approximation of the Bilateral Filter Using a Signal Processing Approach". In: *ECCV (4)*. 2006, pp. 568–580.

[PKG99] Marc Pollefeys, Reinhard Koch, and Luc J. Van Gool. "Self-Calibration and Metric Reconstruction Inspite of Varying and Unknown Intrinsic Camera Parameters". In: *International Journal of Computer Vision* 32.1 (1999), pp. 7–25.

[PKT09] Sylvain Paris, Pierre Kornprobst, and Jack Tumblin. *Bilateral Filtering*. Hanover, MA, USA: Now Publishers Inc., 2009. ISBN: 160198250X, 9781601982506.

[PM90] Pietro Perona and Jitendra Malik. "Scale-Space and Edge Detection Using Anisotropic Diffusion". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 12.7 (1990), pp. 629–639.

[PWS12] Daniele Panozzo, Ofir Weber, and Olga Sorkine. "Robust Image Retargeting via Axis-Aligned Deformation". In: *Computer Graphics Forum (Proceedings of Eurographics)* 31.2pt1 (May 2012), pp. 229–236. ISSN: 0167-7055. DOI: 10.1111/j.1467-8659.2012.03001.x. URL: http://dx.doi.org/10.1111/j.1467-8659.2012.03001.x.

[Red+02] André Redert, Marc Op de Beeck, Christoph Fehn, Wijnand IJsselsteijn, Marc Pollefeys, Luc J. Van Gool, Eyal Ofek, Ian Sexton, and Philip Surman. "ATTEST: Advanced Three-dimensional Television System Technologies". In: *3DPVT*. 2002, pp. 313–319.

[Rei+05] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec. *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann, 2005.

[Rhe+11] Christoph Rhemann, Asmaa Hosni, Michael Bleyer, Carsten Rother, and Margrit Gelautz. "Fast cost-volume filtering for visual correspondence and beyond". In: *CVPR*. IEEE, 2011, pp. 3017–3024.

[RSA08] Michael Rubinstein, Ariel Shamir, and Shai Avidan. "Improved seam carving for video retargeting". In: *ACM Trans. Graph.* 27.3 (2008), p. 16.

[RSA09] Michael Rubinstein, Ariel Shamir, and Shai Avidan. "Multi-operator media retargeting". In: *ACM Trans. Graph.* 28.3 (2009), p. 23.

[Rub+10] Michael Rubinstein, Diego Gutierrez, Olga Sorkine, and Ariel Shamir. "A comparative study of image retargeting". In: *ACM Trans. Graph.* 29.6 (2010), p. 160.

[SA06] M. Segal and K. Akeley. *The OpenGL Graphics System: A Specification (Version 2.1)*. http://www.opengl.org. 2006.

[Sch+12] Lars Schnyder, Manuel Lang, Oliver Wang, and Aljoscha Smolic. "Depth image based compositing for stereo 3D". In: *3DTV-Conference*. 2012, pp. 1–4.

[SD96] Steven Seitz and Charles Dyer. "View Morphing". In: *SIGGRAPH 96*. 1996, pp. 21–30.

[Set+05] Vidya Setlur, Saeko Takagi, Ramesh Raskar, Michael Gleicher, and Bruce Gooch. "Automatic image retargeting". In: *MUM*. 2005, pp. 59–68.

[SH09] G. Sun and N.S. Holliman. "Evaluating methods for controlling depth perception in stereoscopic cinematography". In: *Stereoscopic Displays and Virtual Reality Systems XX, Proceedings of SPIE* 7237 (Jan. 2009).

[Sha+98] Jonathan Shade, Steven J. Gortler, He Li-wei, and Richard Szeliski. "Layered Depth Images". In: *SIGGRAPH*. 1998, pp. 231–242.

[Sin+06] Sudipta N Sinha, Jan-Michael Frahm, Marc Pollefeys, and Yakup Genc. "GPU-based video feature tracking and matching". In: *EDGE, Workshop on Edge Computing Using New Commodity Architectures*. Vol. 278. 2006, p. 4321.

[SLK09] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. "SCRAMSAC: Improving RANSAC's Efficiency with a Spatial Consistency Filter". In: *ICCV*. 2009.

[SLS12] Nikolce Stefanoski, Manuel Lang, and Aljoscha Smolic. "Image quality vs rate optimized coding of warps for view synthesis in 3D video applications". In: *ICIP*. 2012, pp. 1289–1292.

[Smo+08] Aljoscha Smolic, Karsten Müller, Kristina Dix, Philipp Merkle, Peter Kauff, and Thomas Wiegand. "Intermediate View Interpolation Based on Multiview Video Plus Depth for Advanced 3D Video Systems." In: *ICIP*. IEEE, 2008, pp. 2448–2451. URL: `http://dblp.uni-trier.de/db/conf/icip/icip2008.html#SmolicMDMKW08`.

[Smo+10] Aljoscha Smolic, Yongzhe Wang, Nikolce Stefanoski, Manuel Lang, Alexander Hornung, and Markus H. Gross. "Non-linear warping and warp coding for content-adaptive prediction in advanced video coding applications". In: *ICIP*. 2010, pp. 4225–4228.

[Smo+11a] Aljoscha Smolic, Peter Kauff, Sebastian Knorr, Alexander Hornung, Matthias Kunter, Marcus Müller, and Manuel Lang. "Three-Dimensional Video Postproduction and Processing". In: *Proceedings of the IEEE* 99.4 (2011), pp. 607–625.

[Smo+11b] A Smolic, S Poulakos, S Heinzle, P Greisen, M Lang, A Hornung, M Farre, N Stefanoski, O Wang, L Schnyder, et al. "Disparity-aware stereo 3d production tools". In: *Visual Media Production (CVMP), 2011 Conference for*. IEEE. 2011, pp. 165–173.

[SMW06] Scott Schaefer, Travis McPhail, and Joe D. Warren. "Image deformation using moving least squares". In: *ACM Transactions on Graphics* 25.3 (2006), pp. 533–540.

[SN00] Mel Siegel and Shojiro Nagata. "Just Enough Reality: Comfortable 3-D Viewing via Microstereopsis". In: *IEEE Transactions on Circuits and Systems for Video Technology* 10.3 (Apr. 2000), pp. 387–396.

[SS02] Daniel Scharstein and Richard Szeliski. "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms". In: *International Journal of Computer Vision* 47.1-3 (2002), pp. 7–42.

[Ste+00] Lew B. Stelmach, Wa James Tam, Daniel V. Meegan, and André Vincent. "Stereo image quality: effects of mixed spatio-temporal resolution". In: *IEEE Transactions on Circuits and Systems for Video Technology* 10.2 (2000), pp. 188–193.

[Ste+11] Stefanoski, Espinosa, Wang, Lang, Smolic, Bosse, Farre, Muller, Schwarz, Winken, and Wiegand. *Description of 3D Video Coding Technology Proposal by Disney Research Zurich and Fraunhofer HHI*. MPEG, 2011.

[Ste+13] Nikolce Stefanoski, Oliver Wang, Manuel Lang, Pierre Greisen, Simon Heinzle, and Aljoscha Smolic. "Automatic View Synthesis by Image-Domain-Warping". In: *IEEE Transactions on Image Processing* 22.9 (2013), pp. 3329–3341.

[TM98] Carlo Tomasi and Roberto Manduchi. "Bilateral Filtering for Gray and Color Images". In: *ICCV*. 1998, pp. 839–846.

[VJ04] Paul A. Viola and Michael J. Jones. "Robust Real-Time Face Detection". In: *IJCV* 57.2 (2004), pp. 137–154.

[Vol+11] S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer. "Modeling Temporal Coherence for Optical Flow". In: *Proc. 13th International Conference on Computer Vision (ICCV)*. Barcelona: IEEE Computer Society Press, Nov. 2011.

[Wan+04] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612.

[Wan+08] Yu-Shuen Wang, Chiew-Lan Tai, Olga Sorkine, and Tong-Yee Lee. "Optimized scale-and-stretch for image resizing". In: *ACM Trans. Graph.* 27.5 (2008), p. 118.

[Wan+09a] Yu-Shuen Wang, Hongbo Fu, Olga Sorkine, Tong-Yee Lee, and Hans-Peter Seidel. "Motion-Aware Temporal Coherence for Video Resizing". In: *ACM Trans. Graph.* 28.5 (2009).

[Wan+09b] Yu-Shuen Wang, Hongbo Fu, Olga Sorkine, Tong-Yee Lee, and Hans-Peter Seidel. "Motion-aware temporal coherence for video resizing". In: *ACM Transactions on Graphics* 28.5 (2009).

[Wan+10] Yu-Shuen Wang, Hui-Chih Lin, Olga Sorkine, and Tong-Yee Lee. "Motion-based video retargeting with optimized crop-and-warp". In: *ACM Trans. Graph.* 29.4 (2010).

[Wan+11a] Oliver Wang, Manuel Lang, M. Frei, Alexander Hornung, Aljoscha Smolic, and Markus H. Gross. "StereoBrush: Interactive 2D to 3D Conversion Using Discontinuous Warps". In: *SBM*. Ed. by Tracy Hammond and Andrew Nealen. Eurographics Association, 2011, pp. 47–54. ISBN: 978-1-4503-0906-6.

[Wan+11b] Oliver Wang, Manuel Lang, M. Frei, Alexander Hornung, Aljoscha Smolic, and Markus H. Gross. "StereoBrush: Interactive 2D to 3D Conversion UsingDiscontinuous Warps". In: *SBM*. Ed. by Tracy Hammond and Andrew Nealen. Eurographics Association, 2011, pp. 47–54. ISBN: 978-1-4503-0906-6.

[Wan+11c] Yongzhe Wang, Nikolce Stefanoski, Manuel Lang, Alexander Hornung, Aljoscha Smolic, and Markus H. Gross. "Extending SVC by Content-adaptive Spatial Scalability". In: *ICIP*. 2011, pp. 3493–3496.

[Wer+09] Manuel Werlberger, Werner Trobin, Thomas Pock, Andreas Wedel, Daniel Cremers, and Horst Bischof. "Anisotropic Huber-L1 Optical Flow". In: *British Machine Vision Conference (BMVC)*. 2009.

[Wey+07] Tim Weyrich, Jia Deng, Connelly Barnes, Szymon Rusinkiewicz, and Adam Finkelstein. "Digital bas-relief from 3D scenes". In: *ACM Trans. Graph.* 26.3 (2007), p. 32.

[WGC07] Lior Wolf, Moshe Guttmann, and Daniel Cohen-Or. "Non-homogeneous Content-driven Video-retargeting". In: *ICCV 2007. IEEE 11th International Conference on Computer Vision*. Oct. 2007, pp. 1–6. DOI: 10.1109/ICCV.2007.4409010.

[Wil+10] Meindert Onno Wildeboer, Tomohiro Yendo, Mehrdad Panahpour Tehrani, and Masayuki Tanimoto. "A semi-automatic multi-view depth estimation method". In: *Proceedings of SPIE, Visual Communications and Image Processing* 7744 (2010).

[WS08] Chiao Wang and Alexander A. Sawchuk. "Disparity Manipulation for Stereo Images and Video". In: vol. 6803. 1. SPIE, 2008.

[Xia+06] Jiangjian Xiao, Hui Cheng, Harpreet S. Sawhney, Cen Rao, and Michael A. Isnardi. "Bilateral Filtering-Based Optical Flow Estimation with Occlusion Detection". In: *ECCV (1)*. 2006, pp. 211–224.

[YJ88] Seokkwan Yoon and Antony Jameson. "Lower-upper symmetric-Gauss-Seidel method for the Euler and Navier-Stokes equations". In: *AIAA journal* 26.9 (1988), pp. 1025–1026. URL: http://arc.aiaa.org/doi/abs/10.2514/3.10007.

[ZBW11] Henning Zimmer, Andrés Bruhn, and Joachim Weickert. "Optic Flow in Harmony". In: *International Journal of Computer Vision* 93.3 (2011), pp. 368–388.

[ZHM08] Yi-Fei Zhang, Shi-Min Hu, and Ralph R. Martin. "Shrinkability Maps for Content-Aware Video Resizing". In: *Pacific Graphics*. 2008.

[Zit+04] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon A. J. Winder, and Richard Szeliski. "High-quality video view interpolation using a layered representation". In: *ACM Trans. Graph.* 23.3 (2004), pp. 600–608.

[ZMM95] Ramin Zabih, Justin Miller, and Kevin Mai. "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks". In: *ACM Multimedia*. 1995, pp. 189–200.

[Zwi+02] Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. "EWA Splatting". In: *IEEE Trans. Vis. Comput. Graph.* 8.3 (2002), pp. 223–238.