Fall 11-7-2016

Image Stitching: Handling Parallax, Stereopsis, and Video

Fan Zhang
Portland State University

Recommended Citation

Zhang, Fan, "Image Stitching: Handling Parallax, Stereopsis, and Video" (2016). *Dissertations and Theses.* Paper 3286.
https://doi.org/10.15760/etd.3277

Image Stitching: Handling Parallax, Stereopsis, and Video

by

Fan Zhang

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Computer Science

Dissertation Committee:
Feng Liu, Chair
Wu-chi Feng
Melanie Mitchell
Fu Li

Portland State University
2016

ABSTRACT

Panorama stitching increases the field of view in an image by assembling multiple views together. Traditional stitching techniques are proven to be effective only when dealing with parallax-free monocular images. Many challenges that remain unsolved in the stitching research area include how to stitch monocular images with large parallax, how to stitch stereoscopic images to maintain their stereoscopic consistency and original disparity distribution, and how to create panoramic videos with temporally coherent content. To provide more powerful stitching techniques with more universality, we first develop a parallax-tolerant image stitching technique. With the help of it, we then effectively extend the stitching techniques into the stereoscopic image and the video domain to assist users easily making stereoscopic panoramas and video panoramas.

In this dissertation, we first introduce a parallax-tolerant stitching method, which is a local stitching method to stitch monocular images with large parallax. This method is based on the observation that input images do not need to be perfectly aligned over the whole overlapping region for stitching. Instead, they only need to be aligned in a way that there exists a local region where they can be seamlessly blended together. We develop a randomized algorithm to search for a local homography, which, combined with content-preserving warping, allows for optimal stitching. Our experiments show that our method can effectively stitch images with large parallax that are difficult for existing methods.

After studying the problem of regular 2D image stitching, we continue to research 3D image stitching in this dissertation. In particular, we develop a technique for stitching stereoscopic panoramas from stereo images casually taken using a stereo camera. Stereoscopic image stitching needs to address three challenges: how to deal with parallax, how to stitch the left- and right-view panorama consistently, and how to take care of disparity during stitching. We address these challenges by first stitching the left images with the parallax-tolerant image stitching method to create an artifact-free left view panorama, then stitching the disparity maps with disparity optimization, finally warping and stitching the right images according to the stitched disparity map and the left view panorama. Experiment results show that our technique allows for easy production of high quality stereoscopic panoramas that deliver a pleasant stereoscopic 3D viewing experience.

With the 3D image stitching problem addressed, we further study a more complex and challenging task of video stitching. We contribute two video stitching techniques, namely the motion map guided video stitching and the feature trajectory guided video stitching. Our techniques stitch pre-synchronized videos captured from a fixed or hand-held camera array which contains multiple cameras with fixed inter-camera configurations. One unique challenge for video stitching is how to maintain temporal coherence. To address this problem, we propose to consistently stitch frames with the guidance of the target camera motion path. In particular, we develop two techniques using dense motion maps and sparse motion vectors to compute the target camera motion path. Afterwards, we warp and stitch frames according to the target camera motion path to create panoramic videos with temporal coherence. Experiments show that our methods can improve the overall panoramic video stitching quality compared with existing methods.

DEDICATION

To my parents, Hanxiang Zhang and Lianfeng Zhang

To my husband, Yunsong Guo

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Feng Liu, who not only gave me thorough and insightful guidance in the respective research domains with the highest standards, but also entrusted me with tremendous freedom to explore my industrial and research interests. Without his constant support and encouragement, this dissertation would not have been accomplished.

My sincere thanks also goes to Prof. Wu-chi Feng, Prof. Melanie Mitchell, and Prof. Fu Li for serving on my dissertation committee. They have provided continuous help and inspirational advice for my research. Their in-depth knowledge has often accelerated my research and broadened my understanding of the research problems.

I also want to thank all my fellow labmates in the Computer Graphics and Vision Lab: Long Mai, Cuong Nguyen, Hoang Le and Si Lu. Working with them benefited me a lot and made my Ph.D. study more enjoyable.

Finally, and most importantly, I would like to thank my parents for their unconditional and endless love, support, encouragement and inspiration. Special thanks to my beloved husband Dr. Yunsong Guo for his love and care.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

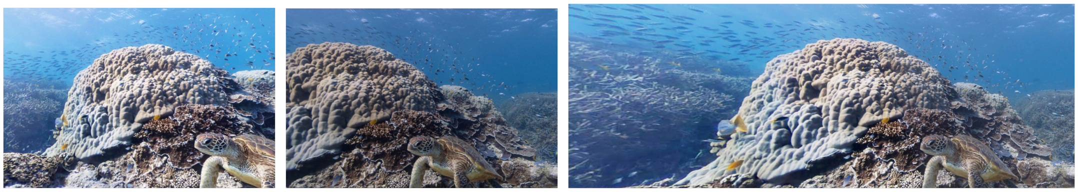
Chapter 1

INTRODUCTION

Stitching techniques merge multiple inputs into a single seamless wider-view output. The inputs can be regular 2D images, stereoscopic 3D images or videos. Regular 2D image stitching is a well-studied topic [53], although the parallax problem remains to be a challenge. Many software tools are available for users, such as AutoStitch [7], Photoshop [1], PTgui [9], and Hugin [6]. On the contrary, stereo stitching and video stitching are still very difficult tasks, and limited research has been done on these two areas. In this chapter, we briefly introduce the motivation and contributions of this dissertation.

1.1 MOTIVATION

Image stitching can create wide field of view panoramas; an example of image stitching can be seen in Figure 1.1. General image stitching techniques usually contain three basic steps, namely registration, alignment and composition. The registration step first estimates feature correspondences among input images. Then the alignment step estimates a 2D transformation between two images and uses the transformation to align input images [7, 55]; this transformation is typically a homography. Finally, these aligned images are composed together to create the final stitching result by performing optimal seam finding, such as seam cutting [3,

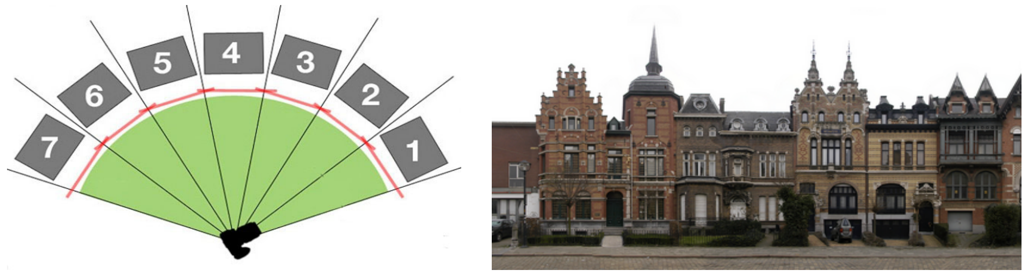


(a) Input image I_1 (b) Input image I_2 (c) Stitching result

Figure 1.1: Image stitching.

28] and blending [8, 39]. These techniques are proven to be effective when dealing with *parallax-free* input images. Parallax is a phenomenon that when you move your viewpoint side to side, the objects in the distance appear to move more slowly than the objects close to you. In order to capture parallax-free images, users need to follow either one of the following two specific shooting rules: all input images should be taken from a fixed viewpoint, as shown in Figure 1.2(a), or the scene of the input images should be roughly planar, as shown in Figure 1.2(b). However, it can be rather challenging to keep the viewpoint fixed when taking multiple images without using a tripod. In addition, scenes with large depth variation are very common. As a result, images casually taken by hand-held cameras usually have parallax, and it is difficult for existing image stitching techniques to perform well in handling these images and creating artifact-free panoramas. Parallax is an intractable problem that haunted researchers for many years, and there is still no effective solution to solve it.

Another problem of existing stitching tools is that they are designed for stitching regular 2D images only and they cannot process stereoscopic 3D images. However, with the increasing popularity of stereoscopic 3D technologies, 3D consumer markets have brought about the demand for stereoscopic image stitching techniques to stitch stereo images and create more immersive 3D image viewing experiences. Simply using existing 2D image stitching techniques to stitch stereoscopic images would create poor results that cause visual fatigue to users. At present, there are



(a) (b)

Figure 1.2: Two cases where existing image stitching tools work well. (a) shows that multiple images should be taken from a fixed viewpoint. (b) shows that the scene of the input images should be roughly planar (the image is take from [2]).

no methods available for effectively stitching multiple stereoscopic images and for creating comfortable 3D stitching results.

In addition to images, stitching techniques can also be applied to videos. Stitching multiple videos together to create 360 degree panoramic videos now is the key component for Virtual Reality applications. Existing techniques either create poor stitching results with noticeable seams on the overlapping area or rely on specific professional camera rigs which are not accessible to amateur users.

1.2 RESEARCH CONTRIBUTIONS

In this dissertation, I thoroughly explore the stitching techniques for regular 2D images, stereoscopic 3D images and videos. My first contribution is a parallax-tolerant image stitching method to handle images with large parallax. With the help of such a method, a stereoscopic image stitching technique is then developed to generate high quality stereoscopic panoramas with stereoscopic consistency and original depth distribution. Finally, I extend image stitching into the video domain and contribute two video stitching techniques to create panoramic videos with temporally coherent content. Each of these areas are briefly introduced below.

Parallax handling. The parallax-tolerant image stitching method is built upon an observation that aligning images perfectly over the whole overlapping area is not necessary for image stitching. Instead, the images only need to be aligned in such a way that there exists a local region in the overlapping area where these images can be stitched together. This stitching strategy is given a term ***local stitching*** and a method is developed accordingly to find such a local alignment that allows for optimal stitching. The local stitching method in this study adopts a hybrid alignment model that uses both homography and content-preserving warping. Homography can preserve global image structures but cannot handle parallax. In contrast, content-preserving warping can better handle parallax than homography, but it cannot preserve global image structures as well as homography. Moreover, local stitching still prefers a well aligned and large local common region. However, when homography is used for aligning images with large parallax, the local region size and alignment quality are often two conflicting goals. This problem is addressed by using homography to only roughly align images and employing content-preserving warping to refine the alignment.

To implement this hybrid local alignment model, a randomized algorithm is firstly developed to search for a homography for inexact local alignment. Therein, a prediction regarding how well the estimated homography enables local stitching can be made by finding a plausible seam from the roughly aligned images and using the seam cost to score the homography. Specifically, a graph-cut based seam finding method is developed to estimate a plausible seam from only roughly aligned images by considering both geometric alignment and image content. Once the optimal homography is determined, it is used for pre-aligning the input images, followed by the content-preserving warping step for refining the alignment. Finally, the well aligned images are composed together to obtain the stitching result.

Stereoscopic image stitching. My second contribution is a stereoscopic image stitching method that enables users to generate stereoscopic panoramas from casually taken stereo images as conveniently as monocular ones. A good stereoscopic stitching method should be able to handle three problems. First, it should handle parallax well. No matter how a user moves a stereo camera, images from at least one of the left and right view have parallax. If users freely move the stereo camera, it is common that images from both views have parallax. Second, the stitching algorithm should stitch the left and right panorama consistently. Third, the algorithm should take care of disparity to deliver a comfortable 3D viewing experience.

To handle the above challenges, a three-step stereoscopic image stitching method is proposed. First, a parallax-tolerant monocular image stitching method is employed to create one of the two views of the stereoscopic panorama. To avoid loss of generality, the left-view panorama is always selected first for stitching. Second, the disparity maps of the input stereoscopic images are stitched together to create the target disparity map for the stereoscopic panorama by solving a Poisson's equation. This target disparity map is optimized to avoid vertical disparities and to seamlessly merge the perceived depth field of the input stereoscopic images. Finally, the right views of the input stereoscopic images are warped and stitched into the right-view panorama according to the target disparity map and the left-view panorama. The stitching of the right views is formulated as a labeling problem that is constrained by the stitching of the left views to make the left- and right-view panorama consistent. The final stereoscopic panorama then can be created by combining the left-view and right-view panorama together.

Video stitching. Finally I contribute two video stitching techniques: the dense motion map guided video stitching and the feature trajectory guided video stitching. Inspired by the stereoscopic image stitching technique, the dense motion map guided video stitching technique is developed to stitch videos with the guidance of the motion field which is estimated from temporally consecutive frames. All video frames are categorized into four types: independent frames, full-reference frames, reduced-reference frames and semi-independent frames. In this way, different stitching methods can be selected according to frame types. Specifically, independent frames do frame stitching independently; full-reference frames do frame stitching based on previous frame stitching result and motion field information; reduced-reference frames do frame stitching based on limited previous frame stitching output and motion field information; and semi-independent frames do frame stitching only based on motion field information. With such a method, stitching errors caused by inaccurate motion field estimation and non-ideal capturing conditions can be properly handled. To further improve video stitching efficiency and accuracy, a video stitching technique based on feature trajectory guidance is then developed. This method generates a desired camera motion path using the in-between middle trajectory of two corresponding feature trajectories. Afterwards, global homography and content-preserving warping are both used for warping individual frames to match with the guidance of the ideal camera motion path. Finally, alpha blending is used in order to blend all warped frames together. This method produces more aesthetically favorable results than the dense motion map guided video stitching technique. Aside from this, without dense motion map estimation, this method is much faster than the first method.

1.3 OUTLINE OF THE DISSERTATION

The structure of this dissertation is organized as follows. In Chapter 2, a brief overview of the background for the rest of the chapters is given, beginning with an introduction of a general 2D image stitching pipeline. The background for stereoscopic 3D image manipulation is also presented, as well as some basic knowledge for video stitching.

Chapter 3 details the parallax-tolerant image stitching technique that handles images with large parallax. The chapter describes the limitations of the existing monocular image stitching method, as well as related work that has already been done to handle these limitations. This is followed by the proposed local stitching method, which uses a hybrid alignment model combining both homograph and content-preserving warping. The chapter also includes experiments conducted to test the stitching algorithm and comparisons of the performances from the proposed algorithm with other state-of-the-art stitching algorithms.

Chapter 4 focuses on the problem of stereoscopic image stitching. The challenges of stereo stitching are firstly described, followed by an introduction of the effective 3-step stereo stitching technique in this study to address the stereo stitching challenges. Afterwards, the chapter gives a detailed description of both a novel disparity stitching algorithm and a seam-cutting method that can maintain stereoscopic consistency. The experiments that were conducted for testing the proposed stereo stitching algorithm are included in the chapter as well.

In Chapter 5, two video stitching techniques are presented. The chapter first introduces the dense motion map guided video stitching method which uses motion map as the guidance to stitch videos with temporal coherence. In order to generate panoramic video in a more reliable and faster way, another feature trajectory based video stitching method is then developed. Finally, the chapter describes the experiments that were conducted using a range of videos to test the performance of the proposed algorithms.

Chapter 6 presents the conclusion of this research; the chapter reviews the contributions of this dissertation, and it also describes some future directions in this research area.

Chapter 2

BACKGROUND

In this chapter, we first briefly introduce the existing monocular image stitching pipeline, and explain in more detail what parallax is and why it is a problem for existing image stitching techniques. Then we introduce the necessary background knowledge for stereoscopic image manipulation. Finally, we conclude this chapter by discussing video stitching basics.

2.1 MONOCULAR IMAGE STITCHING

Researchers have developed a wide range of methods for image stitching [53]. Most of these methods share the same pipeline, as shown in Figure 2.1. This pipeline typically contains three steps, namely registration, alignment, and composition; these three are described in more detail in the section below.

2.1.1 Existing image stitching pipeline

Registration

The first step is to register two input images, and it consists of two sub-steps. First, feature points are detected in each image independently. Afterwards, the feature point correspondences between the two input images are established. This step of image registration has been well studied in computer vision, as reviewed in [53]. Figure 2.2(a) and Figure 2.2(b) show an example.

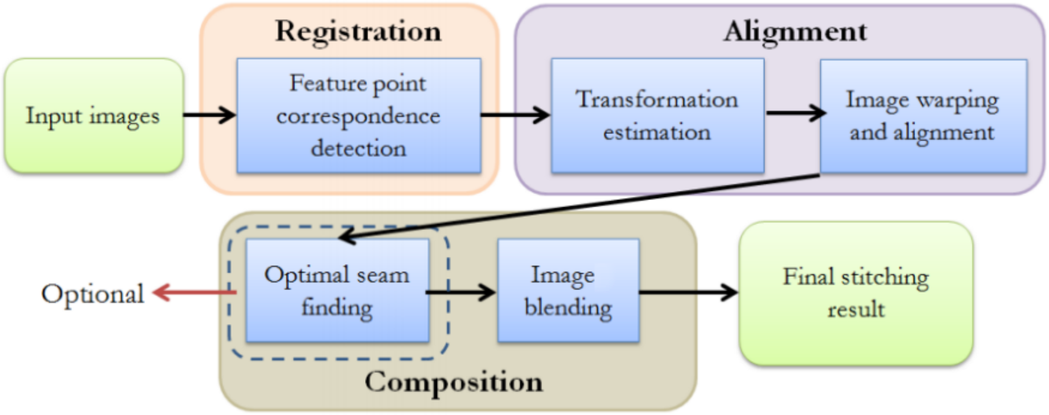


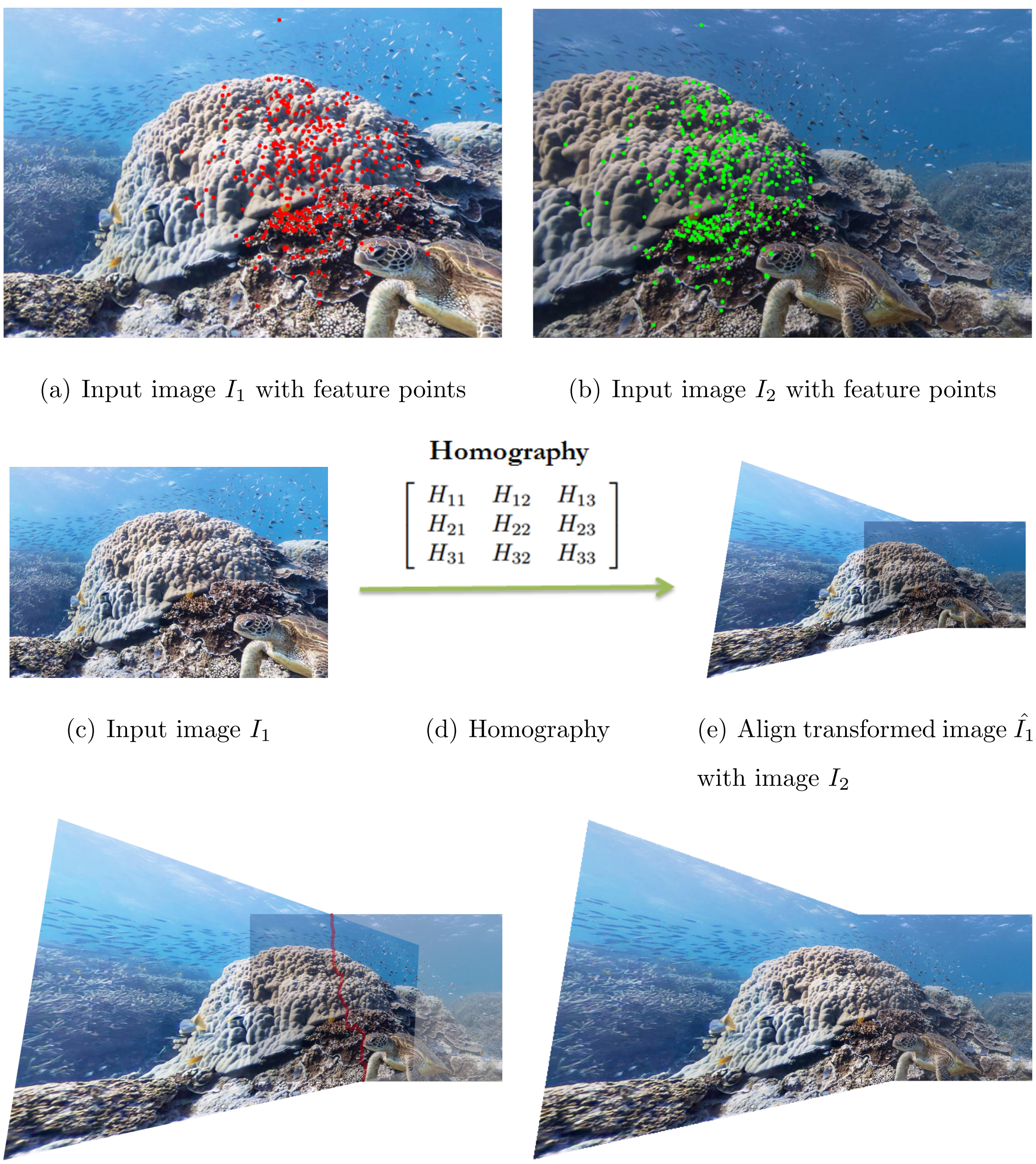
Figure 2.1: Image stitching pipeline.

Alignment

The second step is to estimate a *projective transformation* between two images and to use the transformation to warp and align these images. Existing image stitching techniques assume that there is a projective transformation between two input images, and estimate the projective transformation according to the feature point correspondences. This projective transformation can be represented as a 3×3 transformation matrix, called a *homography*, shown as the matrix in Figure 2.2(d). By using a homography, image I_1 can be transformed to image $\hat{I}_1$ which can be better aligned with image I_2, as shown in Figure 2.2(e).

Composition

After image alignment, the aligned images are composed together to create the final stitching result. There are typically two steps in image composition. The first step is to *find an optimal seam* in the overlapping region of the alignment result so that the pixel difference across the seam is minimal; this is shown as the red curve in Figure 2.2(f). The second step is to *blend* the transformed image $\hat{I}_1$


Homography
$\begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}$

(a) Input image I_1 with feature points

(b) Input image I_2 with feature points

(c) Input image I_1

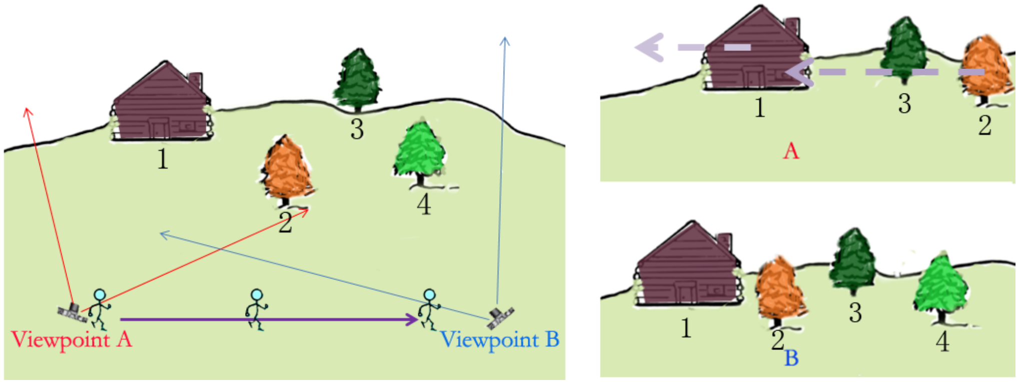
(d) Homography

(e) Align transformed image $\hat{I}_1$ with image I_2

(f) Optimal seam finding result

(g) Blending result

Figure 2.2: Workflow of existing image stitching methods.



(a) Scene with objects in different depth

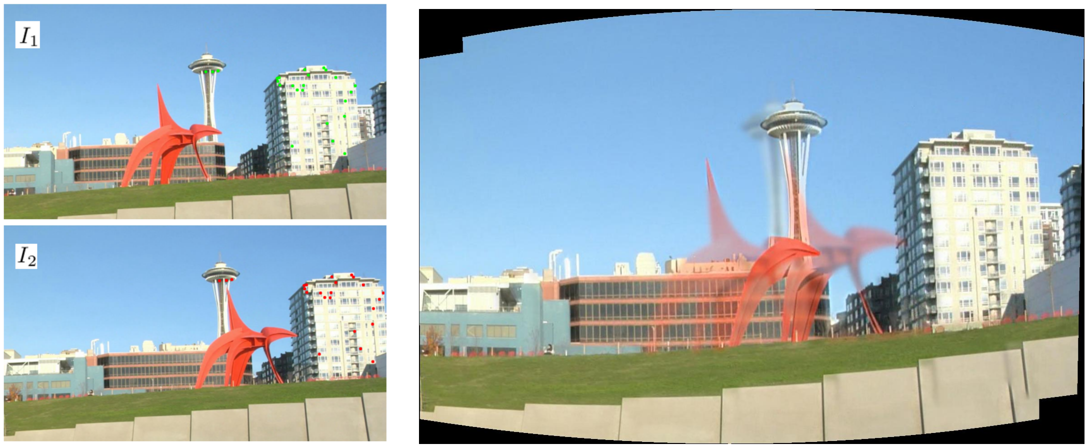
(b) Images taken from viewpoint A and B

Figure 2.3: Images taken from two different viewpoints. (a) shows a scene with objects 1, 2, 3, and 4. Different objects stay in different depth. (b) shows images A and B have parallax. The nearby object 2 has a larger displacement than the distant object 1. And in image A, the objects alignment order is 1 3 2 from left to right, but in image B it becomes 1 2 3.

and image I_2 together to get the final stitching result. Image blending can remove the remaining color and luminance difference along the seam and get the final stitching result, as shown in Figure 2.2(g).

2.1.2 Parallax problem

Existing image stitching techniques assume that the input images can be aligned with a homography-based transformation. However, homography-based transformations cannot be used to account for the parallax between two images. Thus, these methods cannot stitch images with parallax well. Parallax is a phenomenon

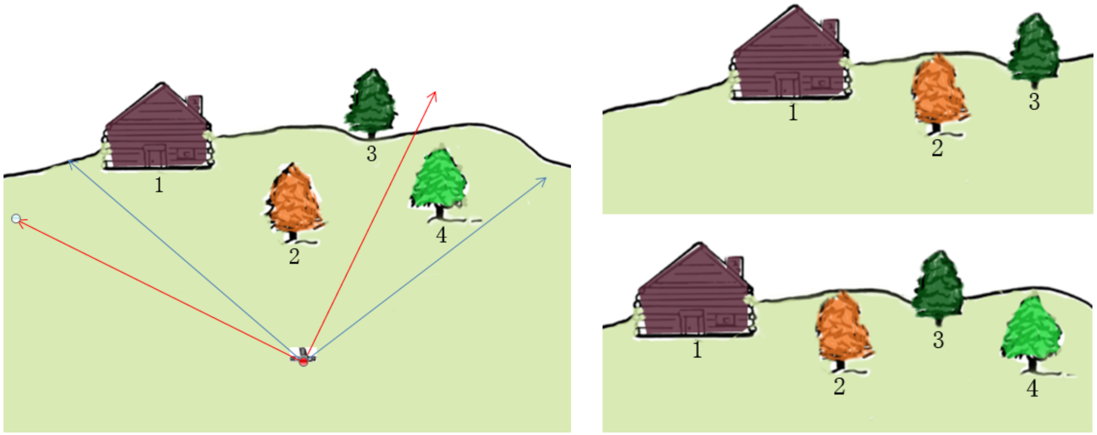


(a) Input image pair

(b) AutoStitch stitching result

Figure 2.4: AutoStitch result.

that when you move your viewpoint side to side, the objects in the distance appear to move more slowly than the objects close to you. According to the multi-viewpoint geometry theory [53], parallax makes the homography-based transformation invalid. As a result, if the input images have parallax, existing image stitching techniques will generate stitching results with artifacts. This limitation can be explained with the following example. Figure 2.3 shows a scene that contains objects 1, 2, 3 and 4; different objects stay in different depths. If we take two images at two different viewpoints as shown in Figure 2.3(a), the nearby object 2 has a larger displacement than the distant object 1 as shown in Figure 2.3(b). In addition, the objects' alignment order is 1, 3, 2 from left to right in image A, but the order becomes 1, 2, 3 in image B. In order to transform image A in a way that it can be matched with image B, the right side of image A needs to be bent so that object 2 can be inserted between object 1 and 3. However, homography-based transformations, which can preserve straight lines in images according to the multi-view geometry theory, cannot bend the image, and thus existing image



(a) Scene with objects in different depth

(b) Images taken from a fixed viewpoint

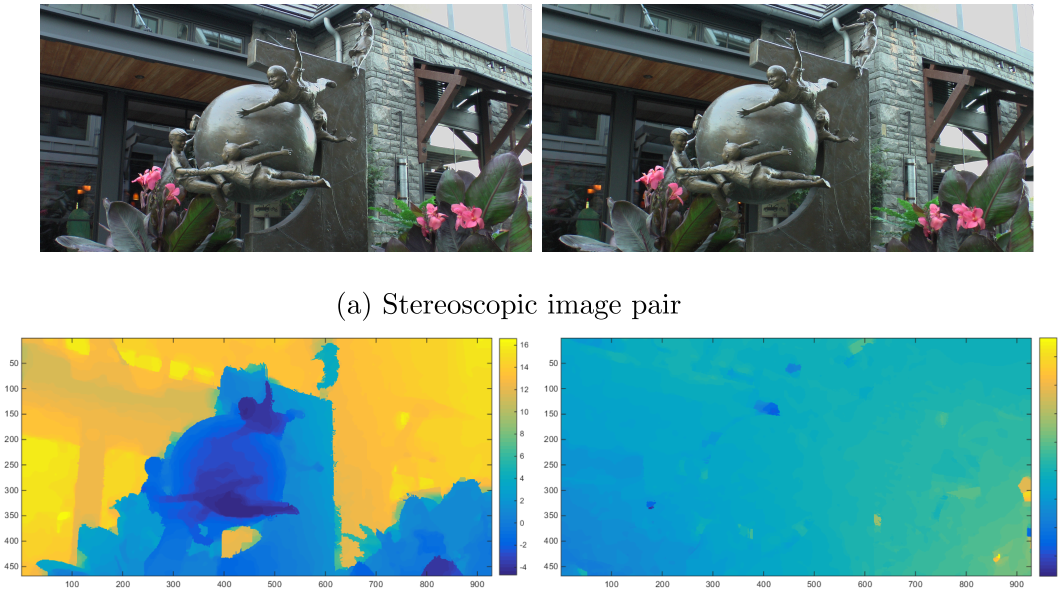
Figure 2.5: Images taken from a fixed viewpoint.

stitching techniques are not able to properly align these two images well aesthetically. As a result, artifacts such as ghosting will be introduced into the final results. Ghosting artifacts are a typical kind of stitching artifacts. Stitching results with ghosting artifacts usually have blurred double imaging for partial image region. Figure 2.4 is an unsuccessful stitching result with ghosting artifacts generated by AutoStitch [7]. As shown in Figure 2.4(a), two images were taken from two different viewpoints. Using a homography-based transformation to transform image I_1 can only match the building in the two images well. As a result, the other regions have ghosting artifacts. However, if we take two images at a fixed viewpoint by rotating the camera as shown in Figure 2.5(a), the two images will not have parallax. For example, the order and displacement of these objects are consistent between the two images in Figure 2.5(b).

2.2 STEREOSCOPIC PHOTOGRAPHY BASICS

Compared to regular 2D images, a stereoscopic 3D image contains two slightly different left and right images of the same scene. The difference between the left and right images provides the information that our brain can use to calculate the depth of the visual scene. This extra dimension of information is called disparity. Specifically, disparity refers to the displacement between the corresponding points in the left and right images. The disparities are usually in two directions: horizontal disparity indicates the horizontal displacement of the corresponding points and vertical disparity indicates the vertical displacement of the corresponding points. Disparity directly affects the perceived distance of an object in a stereoscopic image. If the disparity of an object is zero, it is perceived on the screen. If its disparity is negative, the object pops out of the screen. On the other hand, if the disparity is positive, the object is perceived behind the screen. Figure 2.6 shows an example of the disparity maps of a stereoscopic image. Disparity needs to be taken care of properly and correctly to deliver a comfortable 3D viewing experience that cause no visual fatigue to users. Professionals often adjust the disparities carefully to position the content of interest in the stereoscopic comfort zone [21].

Stereoscopic image editing needs to follow specific editing rules, and the violation of these rules will result in poor stereo images. There are two stereoscopic photography violations that are particularly applicable to stereoscopic image stitching: vertical disparity violation and monocular object violation. Vertical disparities naturally exist in the human visual system. However, when vertical disparity values exceed a certain limit, they will perturb the actual depth perception process and cause visual fatigue [45]. Monocular object violation occurs when salient visual content only shows in one of the two views of a stereoscopic image. This can also produce an uncomfortable viewing experience and can sometimes bring in retinal



(a) Stereoscopic image pair

(b) Disparity maps in horizontal (left) and vertical (right) direction

Figure 2.6: Disparity maps of a stereoscopic image.

rivalry to viewers. For stereoscopic image stitching, stitching the left and right images inconsistently would frequently cause such two stereo violations. Without proper handling, the stereo stitching results cannot deliver a comfortable 3D viewing experience to users.

2.3 VIDEO STITCHING

While image stitching has been extensively studied, there is very limited research done on video stitching topics. In contrast to images, videos have an extra dimension of time. Each point in the frame is the projection of a 3D point in the scene at a specific time. For a fixed 3D point, the position of the projected 2D point in each frame may vary with time. In computer vision, we use motion vectors to represent the displacement of the same point between different frames. Motion vectors of all points in a frame form a motion map (also known as motion field).

Simply extending image stitching into the video domain by stitching each frame separately completely ignores the motion field of the videos, and this would easily introduce temporal incoherence. Temporal incoherence refers to the phenomenon that the temporally adjacent pixels among successive frames deviate from their original motion paths. As a result, temporal incoherence leads to flickering, waving, or ghosting artifacts in the stitched videos. Maintaining temporal coherence is a unique challenge for video stitching.

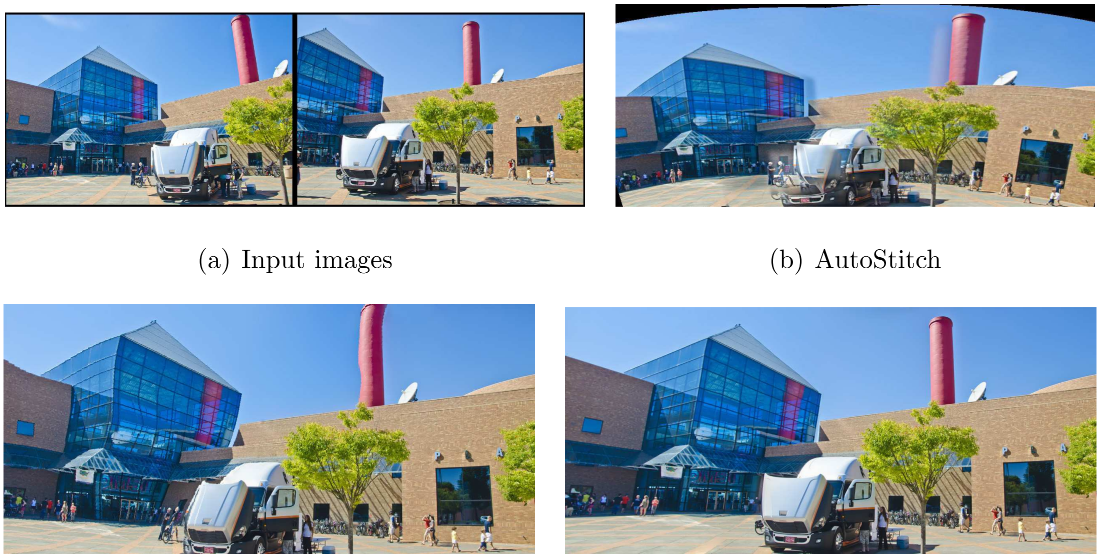
Chapter 3

PARALLAX-TOLERANT IMAGE STITCHING

In this chapter, we present our parallax-tolerant image stitching technique which can handle images with large parallax. We begin by discussing the parallax problem and the limitations of existing stitching methods. We also describe related work that tried to solve the parallax problem. We then present our parallax-tolerant image stitching technique, which is a local stitching method using a hybrid alignment model. We report the performance of our stitching algorithm and also conduct comparison experiments against several state-of-the-art stitching techniques.

3.1 INTRODUCTION

Image stitching is a well-studied topic [53]. Its first step is to register and align input images. Early methods estimate a 2D transformation, typically a homography, between two images and use it to align them [7, 55]. Since a homography cannot account for the parallax between two images, existing image stitching techniques require that the input images should be parallax-free. That is, the input images should be taken from the same viewpoint or the scene should be roughly planar. For images with parallax, no homography exists that can be used to align these images, resulting in artifacts like ghosting or broken image structures. Although advanced image composition techniques can relieve these artifacts, they cannot address significant misalignments.



(a) Input images

(b) AutoStitch

(c) APAP [60]

(d) Our result

Figure 3.1: Parallax problem in monocular image stitching. For images with large parallax, homography-based methods, such as AutoStitch, cannot align input images and introduce ghosting artifacts (b). Spatially-varying warping methods, such as APAP, can align images but introduce apparent visual distortion (c). Our method can produce an artifacts-free result (d)

Recent image stitching methods use spatially-varying warping algorithms to align input images [30, 60]. Spatially-varying warping methods better handle parallax than homography, but they still cannot work well on images with large parallax. Figure 3.1 shows a challenging example with a significant amount of parallax in input images. Notice the horizontal spatial order of the car, the tree, and the chimney in the input images shown in Figure 3.1(a). In the left input image, the chimney is in the middle of the car and the tree while in the right image, the tree is in the middle of the car and the chimney. For this example, one image actually needs to be folded over in order to align with the other. This is a fundamentally difficult task for the warping methods as they either cannot fold over an image or will bring in objectionable distortion, as shown in Figure 3.1(c).

Our parallax-tolerant image stitching method is built upon an observation that aligning images perfectly over the whole overlapping area is not necessary for image stitching. Instead, we only need to align them in such a way that there exists a local region in the overlapping area where these images can be stitched together. We call this ***local stitching*** and develop a method to find such a local alignment that allows for optimal stitching. Our local stitching method adopts a hybrid alignment model that uses both homography and content-preserving warping. Homography can preserve global image structures but cannot handle parallax. In contrast, content-preserving warping can better handle parallax than homography, but cannot preserve global image structures as well as homography. Moreover, local stitching still prefers a well aligned, large local common region. However, when homography is used to align images with large parallax, the local region size and alignment quality are often two conflicting goals. We address this problem by using homography to only roughly align images and employing content-preserving warping to refine the alignment.

We develop a randomized algorithm to search for a homography for inexact local alignment first. Therein, we predict how well a homography enables local stitching by finding a plausible seam from the roughly aligned images and using the seam cost to score the homography. We develop a graph-cut based seam finding method that can estimate a plausible seam from only roughly aligned images by considering both geometric alignment and image content. Once we find the optimal homography, we use it to pre-align the input images and then use content-preserving warping to refine the alignment. We finally compose the well aligned images together to get the stitching result.

3.2 RELATED WORK

Most existing image stitching methods estimate a 2D transformation, typically a homography, between two input images and use it to align them [7, 55]. These homography-based methods can work well only when the input images have little parallax as homography cannot account for parallax. When input images have large parallax, artifacts like ghosting occur. Local warping guided by motion estimation can be used to reduce the ghosting artifacts [49]. Image composition techniques, such as seam cutting [3, 14, 28] and blending [8, 39], have also been employed to reduce the artifacts. However, these methods alone still cannot handle significant parallax. The recent dual-homography warping method can stitch images with parallax, but it requires the scene content can be modeled by two planes [16].

Multi-perspective panorama techniques can handle parallax well [2, 13, 37, 41, 44, 48, 59, 63]. These techniques require 3D reconstruction and/or dense sampling of a scene. They are either time-consuming or cannot work well with only a sparse set of input images, as typically provided by users to make a panorama. The idea behind some of these multi-perspective panorama techniques inspired our work. That is, input images do not need to be perfectly aligned over the whole common image region. As long as we can piece them together in a visually seamless way, an aesthetically favorable artifact-free panoramic image can be created.

A relevant observation has also been made in a recent work that the best-fitting homography does not necessarily enable optimal image stitching [17]. They estimate a set of homographies, each representing a planar structure, create multiple stitching results using these homographies, and find the one with the best stitching quality. This method can successfully handle parallax for some images and also inspired our work; however, it is slow as it needs to create multiple stitching results and evaluate their quality. More importantly, sometimes none of the homographies that represent some planar structures can enable visually plausible stitching.

Recently, spatially-varying warping methods have been extended to image stitching. Lin *et al.* developed a smoothly varying affine stitching method to handle parallax [30]. Zaragoza *et al.* developed a technique to compute an as-projective-as-possible warping that aims to be globally projective while allowing local non-projective deviations to account for parallax [60]. These methods have been shown to work well on images with parallax that are difficult for homography-based methods. However, they still cannot handle images with large parallax, as shown in Figure 4.1.

3.3 ALGORITHM

Our method uses a common image stitching pipeline. Specifically, we first align input images, then use a seam cutting algorithm to find a seam to piece aligned images together [28], and finally employ a multi-band blending algorithm to create the final stitching result [8]. Our contribution is a novel image alignment method which can align images in such a way that allows for optimal image stitching.

Our observation is that we do not need to perfectly align images over their whole overlapping area. In fact, for images with large parallax, it is very difficult, if not impossible, to align them perfectly. Our goal is to align images in a local region where we can find a seam to piece them together. We employ a randomized algorithm to search for a good alignment. We consider an alignment is good enough if it can enable a seamless image stitching. Specifically, we first detect SIFT feature points and match them between two images [32]. We then randomly select a seed feature point and group its neighboring feature points to estimate an alignment as our goal is to estimate an alignment that aligns images over a local region with a compact feature distribution. We evaluate the stitching quality of this alignment. If this alignment is determined good enough to enable a seamless stitching, we stop; otherwise we repeat the alignment estimation and quality evaluation. Below we first discuss some key components of this algorithm and then provide a detailed algorithm description.

3.3.1 Alignment Model Selection

The first question is what alignment model to use. There are two popular options: global 2D transformation, typically homography, and spatially-varying warping, such as content-preserving warping [31, 57]. Most existing methods use a global 2D transformation to align two images. A global 2D transformation has an important advantage in that it warps an image globally and avoids some objectionable local distortions. For example, homography can preserve lines and similarity transformation can preserve the object shape. But they are too rigid to handle parallax. For image stitching, while we argue that it is not necessary to align images exactly in their whole overlapping area, it is still preferable to align images well over an as large as possible common region. However, for images with large parallax, a 2D transformation, even a homography, can often only align images over a small local region. In contrast, content-preserving warping is more flexible and can better align images, but it often introduces objectionable local distortion.

Our solution is to combine these two alignment models to align images well over a large common region with minimal distortion. Given a seed feature point, our method incrementally groups neighboring feature points to fit a 2D transformation (a homography by default). Here we use a slightly large fitness threshold in order to group as many feature points as possible although this makes the homography unable to fit these feature correspondences exactly. Relaxing the fitness threshold of the homography can be compensated by applying content-preserving warping later on, as content-preserving warping is well suited to local warping refinement without introducing noticeable distortion.

3.3.2 Alignment Quality Assessment

A straightforward way to evaluate the stitching quality of the above mentioned hybrid alignment is to first warp an image using the homography and apply content-preserving warping. We can then compare the warped image and the reference image to examine how well these two images are aligned. This approach, however, cannot reliably predict whether a good seam can be found in the overlapping region. Furthermore, this approach does not consider the effect of image content on stitching. For stitching, salient image features, such as edges, should be well aligned while image regions like the sky do not necessarily need to be perfectly aligned. Finally, this approach is slow as it needs to run content-preserving warping whenever we evaluate the alignment quality inside the randomized algorithm.

We address the above problems as follows. First, we examine the alignment quality based on the image edges instead of the raw image directly. Second, we only evaluate how the homography supports stitching. This simplification can be justified by the fact that content-aware warping is very effective if only minor adjustment to the global warping is required. But it also brings in a challenge: the homography in our method is designed to be loose and does not align two images exactly. Then we need to predict how well the alignment enables seamlessly stitching from only roughly aligned images. We address this challenge by finding a plausible seam from the roughly aligned images and using the seam cost to score the alignment.

We first down-sample the input images to both improve speed and tolerate the small misalignment. We then compute the edge maps for the input images using the Canny edge detection method [11]. The edge maps are low-pass filtered to tolerate the small misalignment. We compute the difference between the warped edge map and the reference image's edge map and obtain the difference map E_d. A plausible seam should avoid passing pixels with large values in the difference map in order to obtain a seamless stitching result. We extend the graph-cut seam finding method [28] to find a plausible seam. Briefly, we consider each pixel in the overlapping region as a graph node. We define the edge cost between two neighboring nodes s and t as follows,

$$e(s,t) = f_c(s)|E_d(s)| + f_c(t)|E_d(t)| \tag{3.1}$$

where we use an alignment confidence function $f_c(s)$ to weight the edge cost. $f_c(s)$ is computed to further account for the fact that the homography can only align two images roughly and content-preserving warping will be used to refine the alignment. Specifically, if a local region has a SIFT feature point, the alignment there can very likely be improved by content-preserving warping and thus the misalignment from only using the homography should be deemphasized. We compute $f_c(s)$ to deemphasize the misalignment according to the SIFT distribution as follows,

$$f_c(s) = \frac{1}{\sum_{P_i} g(\|P_s - P_i\|) + \delta} \tag{3.2}$$

where P_i is the position of a SIFT feature point and P_s is the position of pixel s. g is a Gaussian function and is used to propagate the effect of a SIFT feature to its neighborhood. δ is a small constant with a default value 0.01.

Based on the edge cost defined in Equation 3.1, the seam finding problem can be formulated and solved as a graph-cut problem [28]. Once we obtain this seam, we use the cost associated with this seam to score the alignment quality.

Homography Screening

While some homographies can allow for seamless stitching, they sometimes severely distort the images and lead to visually unpleasant stitching results. We detect such homographies and discard them before evaluating their alignment quality. We measure the perspective distortion from applying a homography H to an image I by computing how H deviates from its best-fitting similarity transformation. Denote C_i as one of the four corner points of the input image I and $\bar{C}_i$ is the corresponding point transformed by H. We find the best-fitting similarity transformation $\hat{H}_s$ as follows,

$$\hat{H}_s = \arg\min_{H_s} \sum_{C_i} \|H_s C_i - \bar{C}_i\|^2, \text{where } H_s = \begin{bmatrix} a & -b & c \\ b & a & d \end{bmatrix} \tag{3.3}$$

Once we obtain $\hat{H}_s$, we sum up the distances between the corner points transformed by H and $\hat{H}_s$ to measure the perspective distortion. If the sum of the distances normalized by the image size is larger than a threshold (with default value 0.01), we discard that homography.

3.3.3 Alignment Algorithm Summary

We now describe our randomized algorithm to estimate a good alignment for stitching.

1. Detect and match SIFT features between input images [32] and estimate edge maps for input images [11].

2. Randomly select a valid seed feature point and group its spatially nearest neighbors one by one until the selected feature set cannot be fitted by a homography with a pre-defined threshold. We maintain a penalty value for each feature point to identify the times that it has been selected during the

iteration process. When a feature point is selected, we increase its penalty value by one. In each iteration, to be selected as a seed, a feature point should not have been selected as a seed before and its penalty score is below the average penalty value of all the feature points.

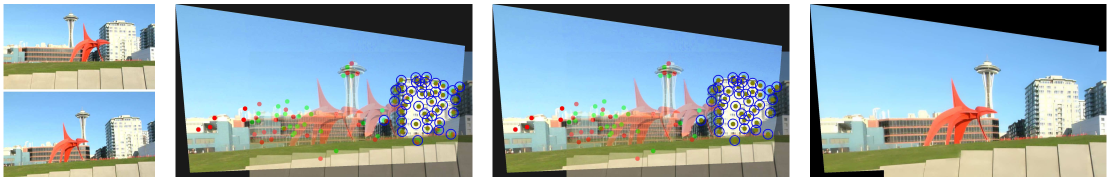
3. Evaluate the alignment quality of the best-fitting homography from Step 2 using the algorithm described in Section 3.3.2. If the homography meets the pre-defined quality threshold, go to Step 4. Otherwise, if the average penalty value is low, go to Step 2; otherwise select the best homography estimated during the iteration process and go to Step 4.

4. Employ the optimal homography to pre-align images and use content-preserving warping guided by the set of selected feature points to refine the alignment, as described in Section 3.3.3.

Figure 3.2 shows the pipeline of our method. Given input images (a), our method first finds an optimal local homography and a subset of feature points that are loosely fit by this homography as shown in (b). We illustrate the selected feature pairs using blue circles. Notice that the homography does not align these features exactly. We then use content preserving warping to refine the alignment. As shown in (c), the selected feature pairs are now well aligned. Our method finally composes the aligned images together (d).

Content-preserving warping

Various content-preserving warping methods have been used in applications, such as video stabilization [31] and image and video retargeting [57, 58]. While content-preserving warping alone cannot always be used to align images over their whole overlapping area, it is well suited for small local adjustment. Therefore, we use



(a) Inputs (b) Optimal local alignment (c) Content-preserving warping (d) Stitching result

Figure 3.2: Stitching pipeline. Please zoom in this figure to better examine the alignment results at (b) and (c). Given input images with large parallax (a), our method first estimates an optimal homography that roughly aligns images locally (b) and is predicted to allow for optimal stitching as described in Section 3.3.2. In (b) and (c), we only blend aligned images by intensity averaging to illustrate alignment. The red and green points are the SIFT feature points in the warped image and the reference image, respectively. When two feature points are aligned, they appear **olive green**. Only a subset of feature points, indicated by blue circles, are selected to fit a homography loosely. Our method then locally refines alignment using content-preserving warping (c), and finally employs seam-cutting and multi-band blending to create the final stitching result (d).

it to further align the pre-warping result from the optimal homography to the reference image as shown in Figure 3.2 (b) and (c).

We use I, $\bar{I}$, and $\hat{I}$ to denote the input image, the pre-warping result, and the final warping result, respectively. We divide the input image I into an $m \times n$ uniform grid mesh. The vertices in I, $\bar{I}$, and $\hat{I}$ are denoted using V_i, $\bar{V}_i$, and $\hat{V}_i$. We then formulate the image warping problem as a mesh warping problem, where the unknowns are $\hat{V}_i$. $\bar{V}_i$ is known from pre-warping. This mesh warping problem is defined as an optimization problem that aims to align $\bar{I}$ to the reference image while avoiding noticeable distortions. We now describe the energy terms in detail below.

Local alignment term. The feature points in image I and $\bar{I}$ should be moved to match their corresponding positions in the reference image so that they can be well aligned. Since a feature point P_j is not usually coincident with any mesh vertex, we find the mesh cell that contains P_j. We then represent $\bar{P}_j$, the corresponding point of P_j in $\bar{I}$, using a linear combination of the four cell vertices of the corresponding cell in image $\bar{I}$. The linear combination coefficients are computed using the inverse bilinear interpolation method [20]. These coefficients are used to combine the vertices in the output image $\hat{I}$ to compute $\hat{P}_j$. We can then define the alignment term as follows.

$$E_p = \sum_{j=1}^{n} \| \sum \alpha_{j,k} \hat{V}_{j,k} - \tilde{P}_j \|^2, \tag{3.4}$$

where n is the size of the selected feature set from the alignment optimization step (Section 3.3.3), $\alpha_{j,k}$ is the bilinear combination coefficient, and $\hat{V}_{j,k}$ is a vertex of the mesh cell that contains $\hat{P}_j$, and $\tilde{P}_j$ is the corresponding feature point in the reference image.

Global alignment term. The alignment term above only directly constrains warping of the overlapping image region with selected feature points. For other regions, content-preserving warping often distorts them. As the pre-warping result $\bar{I}$ has already provided a good approximation, our method encourages the regions without feature points to be close to the pre-warping result as much as possible. We therefore define the following global alignment term,

$$E_g = \sum_{i} \tau_i \| \hat{V}_i - \bar{V}_i \|^2, \tag{3.5}$$

where $\hat{V}_i$ and $\bar{V}_i$ are the corresponding vertex in the content-preserving warping result and in the pre-warping result. τ_i is a binary value. We set it 1 if there is no feature point in the neighborhood of V_i; otherwise it is 0. This use of τ_i provides flexibility for local alignment.

Smoothness term. To further minimize the local distortion during warping, we encourage each mesh cell in the pre-warping result to undergo a similarity transformation. We use the quadratic energy term from [23] to encode the similarity transformation constraint. Specifically, consider a triangle $\triangle \bar{V}_1 \bar{V}_2 \bar{V}_3$. Its vertex $\bar{V}_1$ can be represented by the other two vertices as follows,

$$\bar{V}_1 = \bar{V}_2 + u(\bar{V}_3 - \bar{V}_2) + vR(\bar{V}_3 - \bar{V}_2), R = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \tag{3.6}$$

where u and v are the coordinates of $\bar{V}_1$ in the local coordinate system defined by $\bar{V}_2$ and $\bar{V}_3$. If this triangle undergoes a similarity transformation, its coordinates in the local coordinate system will not be changed. Therefore, the similarity transformation term can be defined as follows,

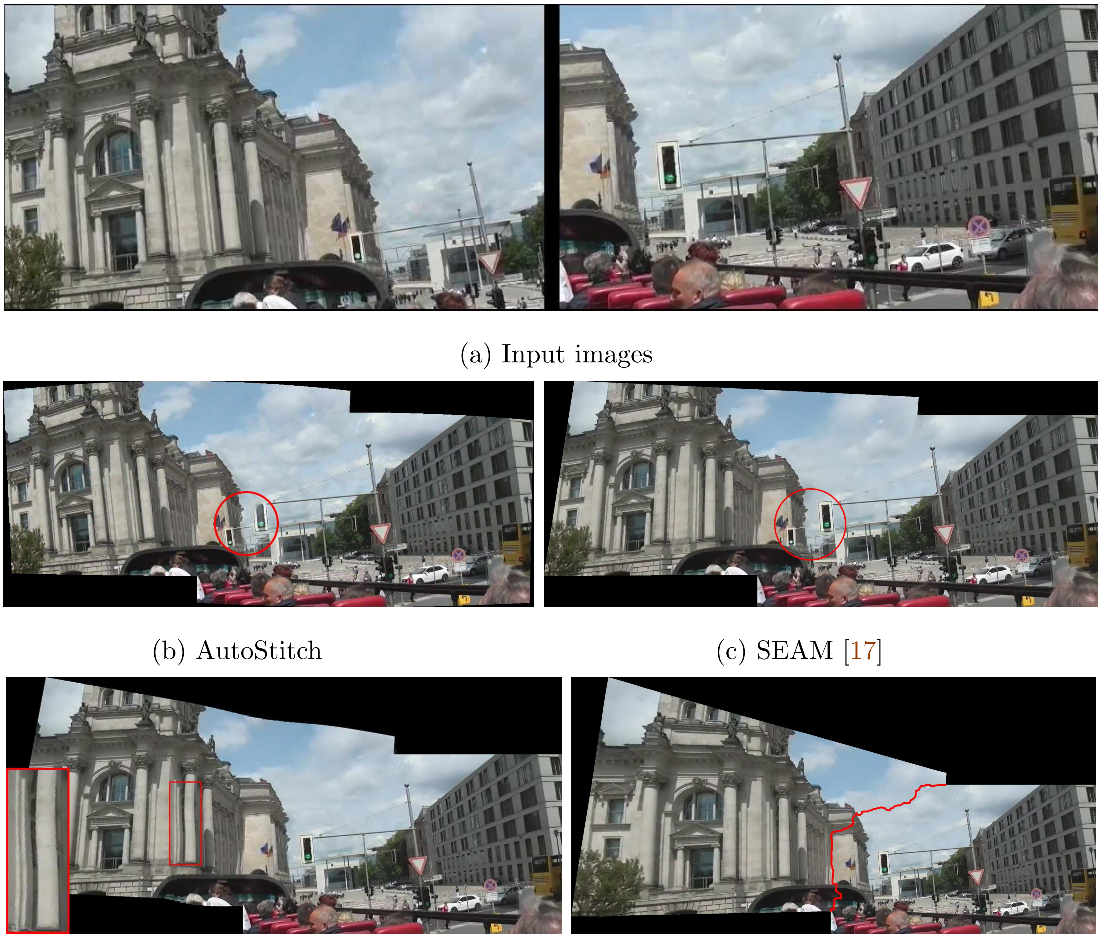
$$E_s(\hat{V}_i) = w_s \| \hat{V}_1 - (\hat{V}_2 + u(\hat{V}_3 - \hat{V}_2) + vR(\hat{V}_3 - \hat{V}_2)) \|^2, \tag{3.7}$$

where u and v are computed from Equation 5.5. We sum $E_s(\hat{V}_i)$ over all the vertices and obtain the full smoothness energy term E_s. Here w_s measures the saliency value of the triangle $\triangle \bar{V}_1 \bar{V}_2 \bar{V}_3$ using the same method as [31]. We use this saliency weight to distribute more distortion to less salient regions than those salient ones.

Optimization. We combine the above three energy terms into the following energy minimization problem,

$$E = \alpha E_p + \beta E_g + \gamma E_s, \tag{3.8}$$

where α, β, γ are the weight of each term with default values 1.0, 0.01, and 0.001, respectively. The above minimization problem is quadratic and is solved using a standard sparse linear solver. Once we obtain the output mesh, we use texture mapping to render the final result.



(a) Input images

(b) AutoStitch

(c) SEAM [17]

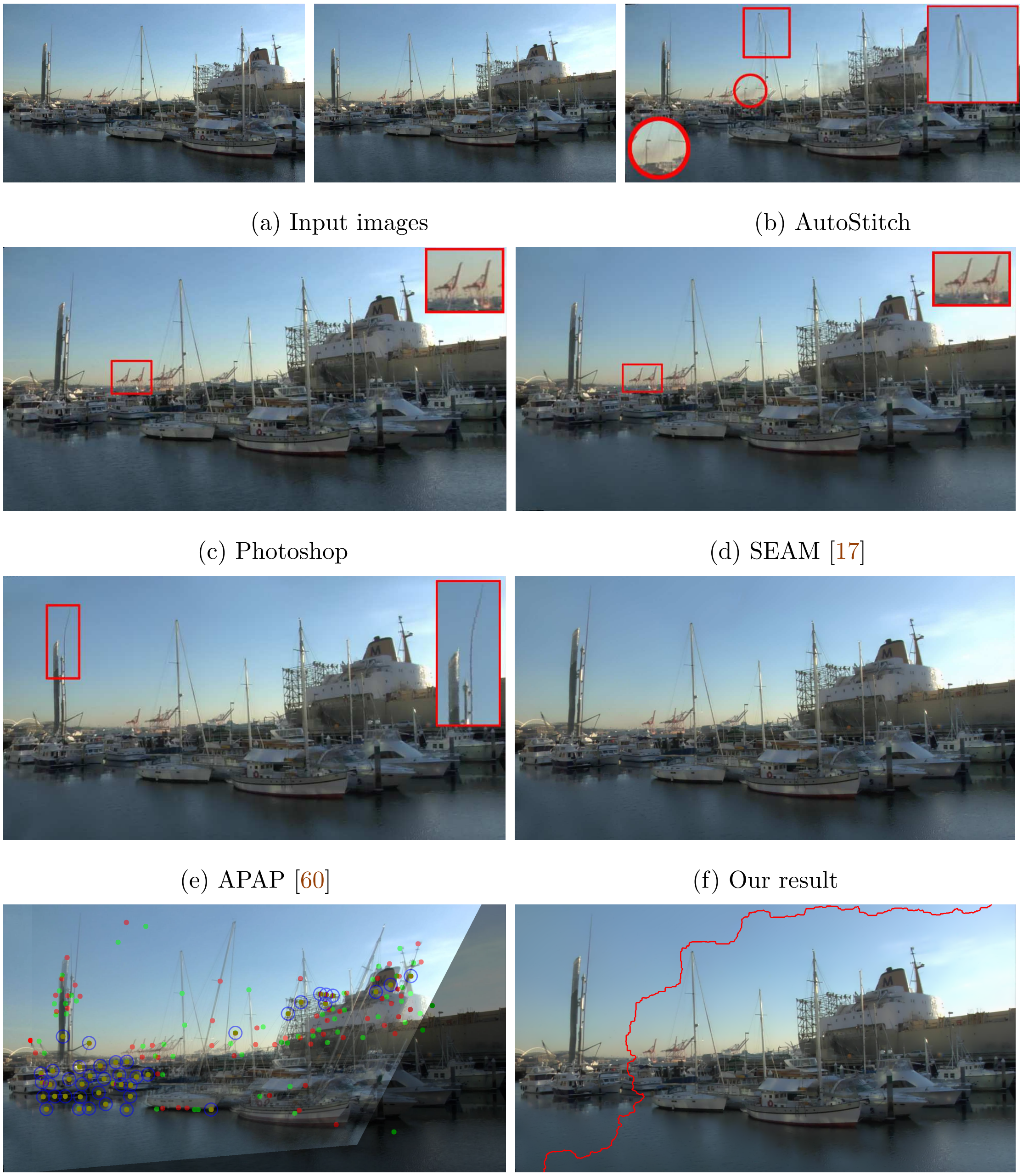
(d) APAP [60]

(e) Our result with seam

Figure 3.3: Comparisons among various stitching methods.

3.4 EXPERIMENTS

We experimented with our method on a range of challenging images with large parallax. We also compared our method to the state-of-the-art methods, including Photoshop, AutoStitch, as-projective-as-possible stitching (APAP) [60], and our implementation of seam-driven stitching (SEAM) [17]. For APAP, we used the code shared by the authors. Since that code only aligns images, we applied the same seam-cutting and multi-band blending algorithm used in our method to the APAP alignment results to produce the final stitching results.



(a) Input images

(b) AutoStitch

(c) Photoshop

(d) SEAM [17]

(e) APAP [60]

(f) Our result

(g) Our result with matching points

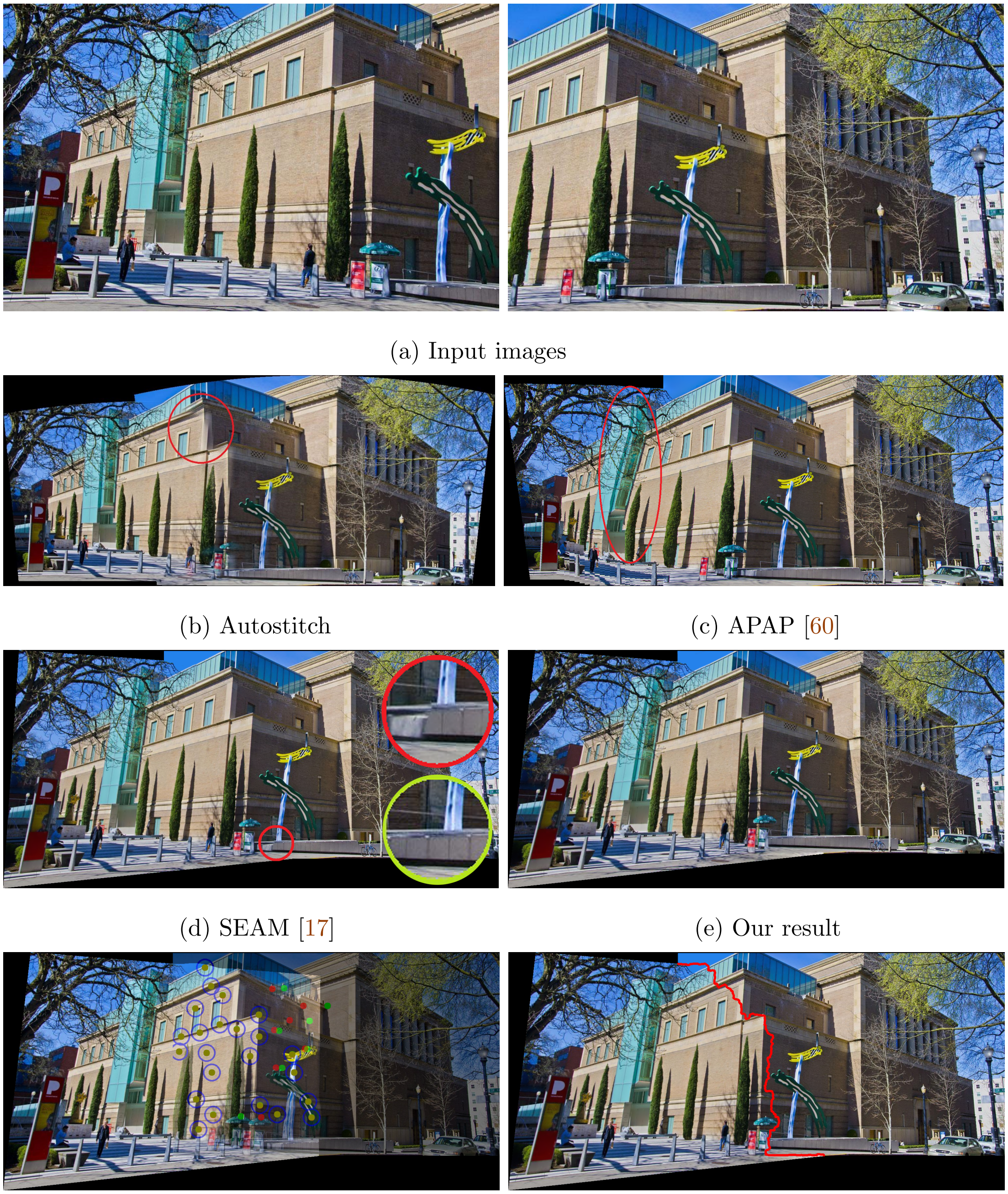
(h) Our result with seam

Figure 3.4: Comparisons among various stitching methods.

Figure 3.3(a) shows two input images with a significant amount of parallax. Photoshop failed to produce any result. AutoStitch could not align two images well using a global 2D transformation, therefore the stitching result suffers from ghosting, as indicated by the red circle in Figure 3.3(b). The traffic light is duplicated in the final result. The SEAM method did not find a local plane represented by a homography that allows for seamless stitching, and duplicated the traffic light too as shown in Figure 3.3(c). The APAP method creates a reasonable stitching result as shown in Figure 3.3(d); however, as APAP tries to align two images over the whole overlapping region, it distorts the salient image structure, such as the pillar indicated by the red rectangle. Our method can handle this challenging example by aligning the input images locally in a way that allows for optimal stitching, as shown in Figure 3.3(e). We also show the stitching seam in red.

Figure 3.4(a) shows another challenging example. The two input images have a large amount of parallax, and there is no global transformation that can align them well over the whole overlapping region. As shown in Figure 3.4(b), the AutoStitch result suffers from significant ghosting artifacts. While blending can relieve misalignment, it causes severe blurring artifacts as indicated by the red circle. Both Photoshop and SEAM duplicated the red structure, as shown in Figure 3.4(c) and Figure 3.4(d). APAP bends the straight line as shown in Figure 3.4(e). Our result in Figure 3.4(f) is free from these artifacts. Detailed alignment and seam finding figures are shown in Figure 3.4(g) and Figure 3.4(h). By finding a local alignment and refine the alignment with content-preserving warping, only part of the image has been well-aligned as shown in Figure 3.4(g), but the well-aligned overlapping region is already good enough to successfully find a plausible seam goes through the well-aligned region and the non-salient region.

Figure 3.5(a) shows another challenging example. There is a large amount of



(a) Input images

(b) Autostitch

(c) APAP [60]

(d) SEAM [17]

(e) Our result

(f) Our result with matching points

(g) Our result with seam

Figure 3.5: Comparisons among various stitching methods.

	Autostitch	Photoshop	APAP	SEAM	Ours
mean	2.41	2.89	2.59	2.98	3.84
std	1.14	1.26	1.15	1.27	1.09
best	1	1	1	2	5

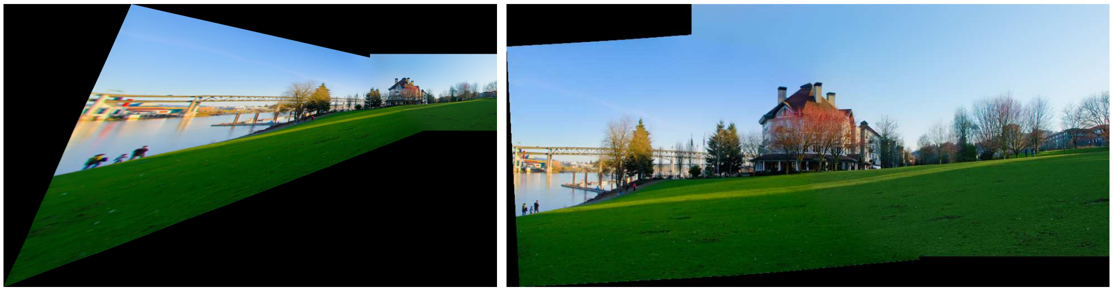
Table 3.1: User study results.

parallax in the input images. No global transformation can align them well over the whole overlapping region. Figure 3.5(b) shows the AutoStitch result, which suffers from significant ghosting artifacts. Photoshop failed to generate any result for this example. APAP significantly bends the straight lines as shown in Figure 3.5(c) and SEAM cannot well align the structure, causes a broken structure, as shown in Figure 3.5(d). Our result in Figure 3.5(e) is free from these artifacts. Figure 3.5(f) and Figure 3.5(g) show the detailed figures for this example. As shown in images, only a local region in the overlapping area is well aligned and a large part of the overlapping region is mismatched. But as long as the seam finding algorithm can find a seam to enable a seamless image stitching, such as the seam in Figure 3.5(g), a local alignment is good enough to generate an artifact-free stitching result.

3.4.1 User study

We also designed a user study to subjectively evaluate the user experience of viewing panoramas generated by our method and four state-of-the-art stitching methods including Photoshop, Autostitch, as-projective-as-possible stitching (APAP) [60], and our implementation of seam-driven stitching (SEAM) [17]. We collected 10 input image pairs and generated five panoramas for each image pair using five stitching methods. We obtained 50 panoramas in total.

There are 20 participants in our study, including 7 females and 13 males. They are students and employees covering a wide variety of background, including Computer Science, Economics, Education, English, Mathematics, Physics, Psychology etc. Their ages range from around 20 to 60 years old. They do not know how each panorama was created. Before the study, we provided five panoramas generated using five different methods for them to look at and to learn what typical stitching artifacts are. In our study, we showed five panoramas generated using different methods for each image pair to each participant one by one in a random order. Participants can look at a panorama as long as they want. After a participant finished looking at a panorama, we asked the participant to rate the panorama stitching quality using a Likert scale ranging from 1 to 5, with 5 being the most positive. After a participant finished looking at all five panoramas generated for one image pair, we asked the participant to choose the image stitching result with the best quality. For each image pair, we consider the most selected stitching result as the best stitching result. We report the average scores, the standard deviations and the number of times each stitching method has been voted to be the best stitching method among the five in Table 3.1. Our study confirms our hypothesis that existing stitching techniques are problematic in handling images with large parallax and will damage the panorama viewing experience. The best average quality score of the four existing methods is 2.98. In contrast, our method produces better viewing experiences with an average score of 3.84. The p-values of the paired two sample t-test between our results and each state-of-the-art results are smaller than 0.001, which shows that the difference between the two sets of results are statistically significant. Panoramas produced by our method have also been selected as the best stitching results for 5 times out of 10 image sets, compared with 2 times for the second best method. In conclusion, the user study clearly shows that our parallax-tolerant image stitching technique generates better



(a) Homography

(b) Similarity transformation

Figure 3.6: Homography vs. Similarity transformation. Our method is flexible in choosing a 2D global transformation for initial alignment. Sometimes we can replace the commonly used homography with a similarity transformation to reduce distortion.

stitching results than other state-of-the-art algorithms.

3.4.2 Discussion

Our method only needs to align input images locally and fit a homography loosely, as described in Section 3.3.2. Therefore our method can sometimes use a more restrictive global transformation than homography to remove the perspective distortion from homography. Figure 3.6(a) shows a stitching result from our method using homography for initial alignment, which suffers from noticeable perspective distortion. Once we replace homography with similarity transformation for initial alignment, the stitching result suffers from less distortion, as shown in Figure 3.6(b). Our technique can also stitch more than two images. Figure 3.7 shows a result created by stitching five images together.

We also tested how the homographies selected by our method differ from the best-fitting ones by computing the distances between the transformed image corner positions with our homographies and the best-fitting ones. Over 75% of the examples shared in our project website has the average corner position distance larger

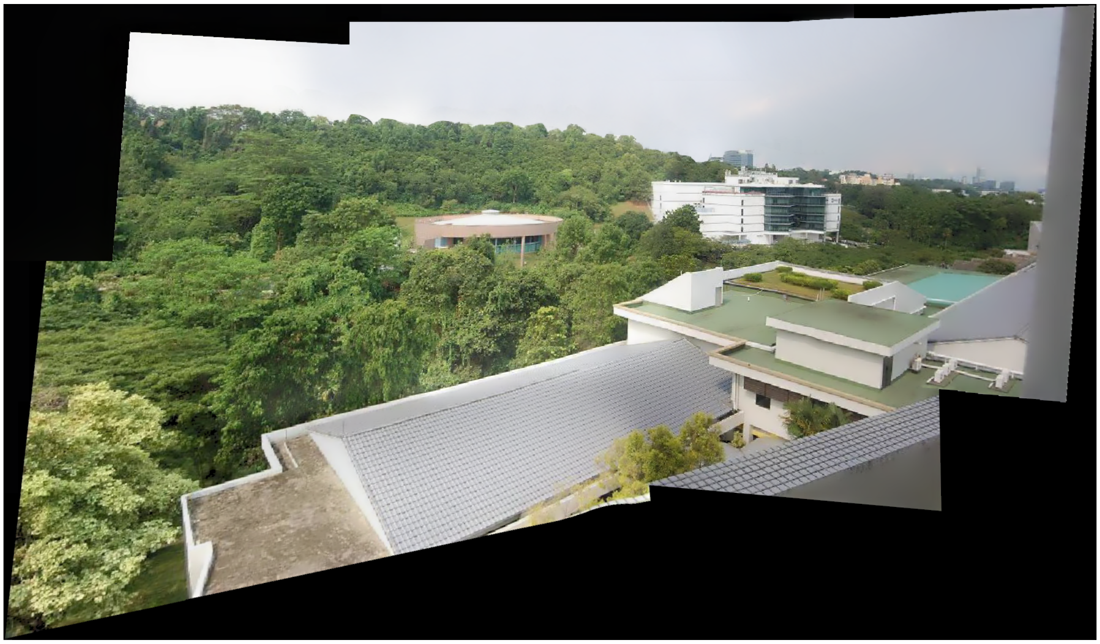
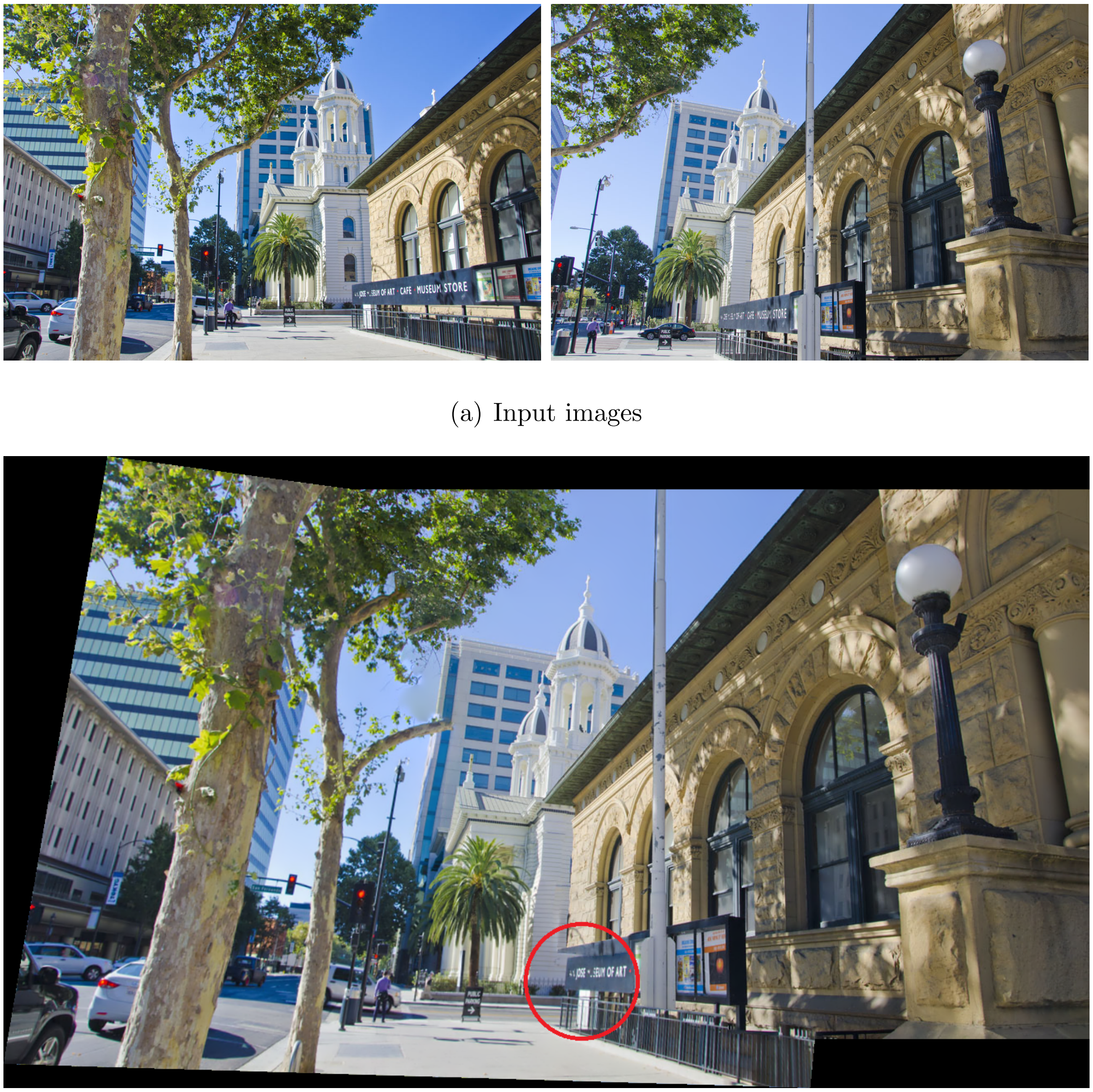


Figure 3.7: Multiple image stitching.

than 36 pixels (given an image with width 1000 pixels) . The median distance is around 60 pixels. This confirms that our method uses different homographies than the best-fitting ones.

Our method works well on a range of examples with large parallax as well as all the examples reported in the recent APAP paper [60]. Meanwhile, we also found some failure cases. Figure 3.8 shows one failure example. The input images have very large parallax and are full of salient structures. For stitching, images must be aligned so that there at least exists a local common region where a good seam can be found. In images with large parallax, there is often no such a local region that can be aligned. Our method explores the fact that non-salient areas often need not be well aligned and considers this in searching for a good local region alignment. But if an image has large parallax and is full of salient structures, our method sometimes cannot work as no non-salient region exists. Our method adopts a common image stitching pipeline. Its major novelty is in its step to align images



(a) Input images

(b) Our result

Figure 3.8: Failure example.

such that optimal stitching can be achieved. This step, including optimal local homography estimation and content-preserving warping, typically takes from 20 to 40 seconds on a desktop machine with Intel i7 CPU and 8 GB memory to align two images with width 1000 pixels. All the other steps are shared by off-the-shelf image stitching methods.

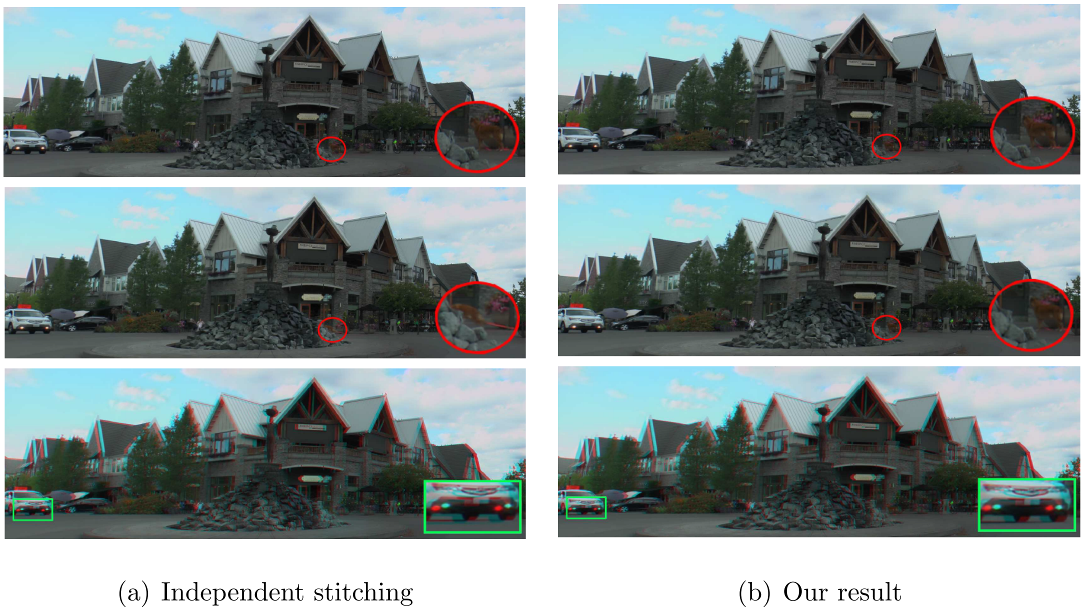
Chapter 4

CASUAL STEREOSCOPIC IMAGE STITCHING

In this chapter, we introduce our casual stereoscopic image stitching technique. We first discuss the challenges of the stereo image stitching task. We also introduce related work in the stereo stitching and parallax handling research areas. Then we present our 3-step casual stereoscopic image stitching technique. Finally, we conduct several experiments to test the performance of our algorithm. We conclude this chapter by discussing the limitations of our method.

4.1 INTRODUCTION

Panorama stitching is a well studied topic and many software tools are available for users to create panoramas [54]. Most of these methods, however, are designed for monocular image stitching. Employing a monocular image stitching method to independently create the left and right view of a stereoscopic panorama is problematic as the left and right panorama may not be consistent. As shown in Figure 4.1(a), the cat in the left panorama is different from that in the right panorama. This is because the input images are taken at different time and the cat appears different in the input images. The left and right panorama take the cat from different input images. The inconsistency will lead to "retinal rivalry" and bring in "3D fatigue" to viewers [33]. Moreover, stereoscopic images have an extra dimension of disparity, which cannot be taken care of by independently stitching the two views. Figure 4.1(a) shows that the resulting panorama has


(a) Independent stitching

(b) Our result

Figure 4.1: Stereoscopic panorama stitching. For each column, we show the left-view, right-view and red-cyan anaglyph of the stereo panorama. Stitching the left- and right-view panorama independently brings in inconsistency artifacts like monocular object (the cat) and vertical disparities (the car headlights), which will cause "3D fatigue" to viewers. Our result is free from these artifacts.

vertical disparities in the car headlight area. This will also compromise the 3D viewing experience of viewers. Dedicated stitching methods have been developed for stereoscopic panorama stitching [22, 36, 43]. However, these methods require a user to densely sample the scene using a video camera and/or follow some specific rules to rotate the camera and cannot work well with a sparse set of casually taken input images.

In this chapter, our research objective is to develop a technology that allows users to create stereoscopic panoramas as conveniently as monocular ones. As consumer stereo cameras now become more and more available to daily users, it becomes easy for them to take stereoscopic images. We therefore aim to develop a stereoscopic image stitching method that enables users to generate stereoscopic panoramas from casually taken stereo images. To achieve this goal, we need to address three challenges. First, our method needs to handle parallax well. No matter how a user moves a stereo camera, images from at least one of the left and right view have parallax. As we allow users to freely move the stereo camera, it is common that images from both views have parallax. Second, our method needs to stitch the left and right panorama consistently. Third, our method needs to take care of disparity to deliver a pleasant viewing experience.

We present a three-step stereoscopic image stitching method to address the above challenges. First, we employ a state-of-the-art parallax-tolerant monocular image stitching method to create one of the two views of the stereoscopic panorama. Without loss of generality, we always select the left-view panorama to stitch first. Second, we stitch the disparity maps of the input stereoscopic images to create the target disparity map for the stereoscopic panorama by solving a Poisson's equation. This target disparity map is optimized to avoid vertical disparities and seamlessly merge the perceived depth field of the input stereoscopic images. Finally, we warp the right views of the input stereoscopic images and stitch them into the right-view panorama according to the target disparity map. The stitching of the right views is formulated as a labeling problem that is constrained by the stitching of the left views to make the left- and right-view panorama consistent.

Our main contribution of this chapter is a stereoscopic image stitching method that allows users to generate stereoscopic panoramas as conveniently as they generate monocular ones. To develop this stereoscopic image stitching method, we also provide a novel algorithm to seamlessly stitch input disparity maps and a seam-cutting method to stitch the right panorama that is consistent with the stitching of the left panorama and respects the target disparity map. Our experiments show our method allows for easy production of stereoscopic panoramas that deliver a pleasant 3D panoramic viewing experience.

4.2 RELATED WORK

Monocular image stitching is a well studied topic. A good survey can be found in [54]. This section focuses on stereoscopic image stitching and techniques for parallax handling which are most relevant to our work.

Stereoscopic image stitching. Stereoscopic panoramas require source images for the left and right panorama to be taken from different viewpoints. These images can be recorded using either a stereo camera or a moving monocular camera [12, 22, 24, 36, 43, 47, 50]. The early *PSI* system uses a stereo camera rig and rotates it horizontally around an axis passing through the optical center of the right camera to collect a set of left images and a set of right images [22]. The left and right panorama are then created using a disparity warping technique and a hierarchical seaming algorithm. Couture *et al.* [12] developed a stereoscopic panoramic video stitching method that captures input videos by rotating a stereo camera rig around an off camera center vertical axis. Peleg *et al.* [36] developed an *omnistereo* panorama system that mounts a monocular camera on a rotating arm to capture images from various viewpoints. The left and right panorama can then be synthesized by taking proper strips from input views. Richardt *et al.* [43] further improved this *omnistereo* method by correcting the deviations from the ideal capture setup and addressing the insufficient sampling problem using a flow-based ray upsampling algorithm. All these existing methods require users to densely sample the scene using a camera and/or follow some specific rules to rotate it. In contrast, our work only requires sparse samples of the scene casually captured by a stereo camera.

Parallax handling. Traditional homography-based image stitching methods cannot handle parallax well [7, 54]. Thus, techniques like local warping [49], seam cutting [3, 14, 28], and blending [8, 39], are developed to reduce or eliminate artifacts caused by parallax. Spatially-varying warps are recently employed to align images for image stitching [30, 60, 61]. Since these methods are more flexible, they can often better handle parallax than homography. Recent research shows that images do not need to be globally aligned to produce a good stitching result. A recent method, instead of estimating a best-fitting homography, searches for a good homography that enables optimal stitching to align input images [17]. A local stitching method further develops this idea and finds a local alignment that combines homography and spatially-varying warp to better handle parallax and allows for optimal stitching [61]. Our method builds upon these existing methods to handle parallax. The first step of our approach uses the recent local stitching method [61] to stitch the left view of the final stereoscopic panorama. Our method also extends a spatially-varying warping method to transform the right views of the input stereoscopic images according to the target disparity map in a way that is robust against parallax.

4.3 ALGORITHM

In this section, we present our casual stereoscopic image stitching technique. Our method takes as input a sparse set of stereoscopic images casually captured using a stereoscopic camera and outputs a stereo panorama. We consider that a good stereoscopic panorama has the following properties. First, both the left and right panorama should be artifact-free. Second, the left and right panorama should be consistently stitched to avoid “retinal rivalry”. Third, the disparity map of the stereoscopic panorama should be carefully taken care of. Stitching should introduce no vertical disparities. Moreover, the horizontal disparity maps of input images should be seamlessly stitched to ensure proper depth perception.

In order to create such a good stereoscopic panorama, our method decomposes stereoscopic panorama stitching into three separate steps after a pre-processing step to estimate disparity maps of input stereoscopic images.

1. Stitch the left panorama from the left views of input stereoscopic images using a state-of-the-art monocular stitching algorithm.

2. Stitch the target disparity map of the output stereoscopic panorama from the disparity maps of input stereoscopic images.

3. Warp the right views of input stereoscopic images and stitch the right panorama according to the stitching of the left panorama and the target disparity map.

For simplicity, we consider the task of stitching two input stereoscopic images I_1 and I_2. More images can be stitched similarly. Each stereoscopic image has a left and right image. For example, $I_{1,l}$ and $I_{1,r}$ are the left and right image of I_1, respectively. We denote the two views of the output stereoscopic panorama as $\hat{I}_l^p$ and $\hat{I}_r^p$.

Our method pre-processes input stereoscopic images to estimate their disparity maps. Like previous methods in stereoscopic image editing [29], we downsample each input stereoscopic image, estimate dense correspondences from the downsampled images using an optical flow method [52], and scale up the resulting optical flow vectors as the disparities of the original image. We denote the disparity map for the input stereoscopic images I_1 and I_2 as D_1 and D_2, respectively. We describe each step of our method below.

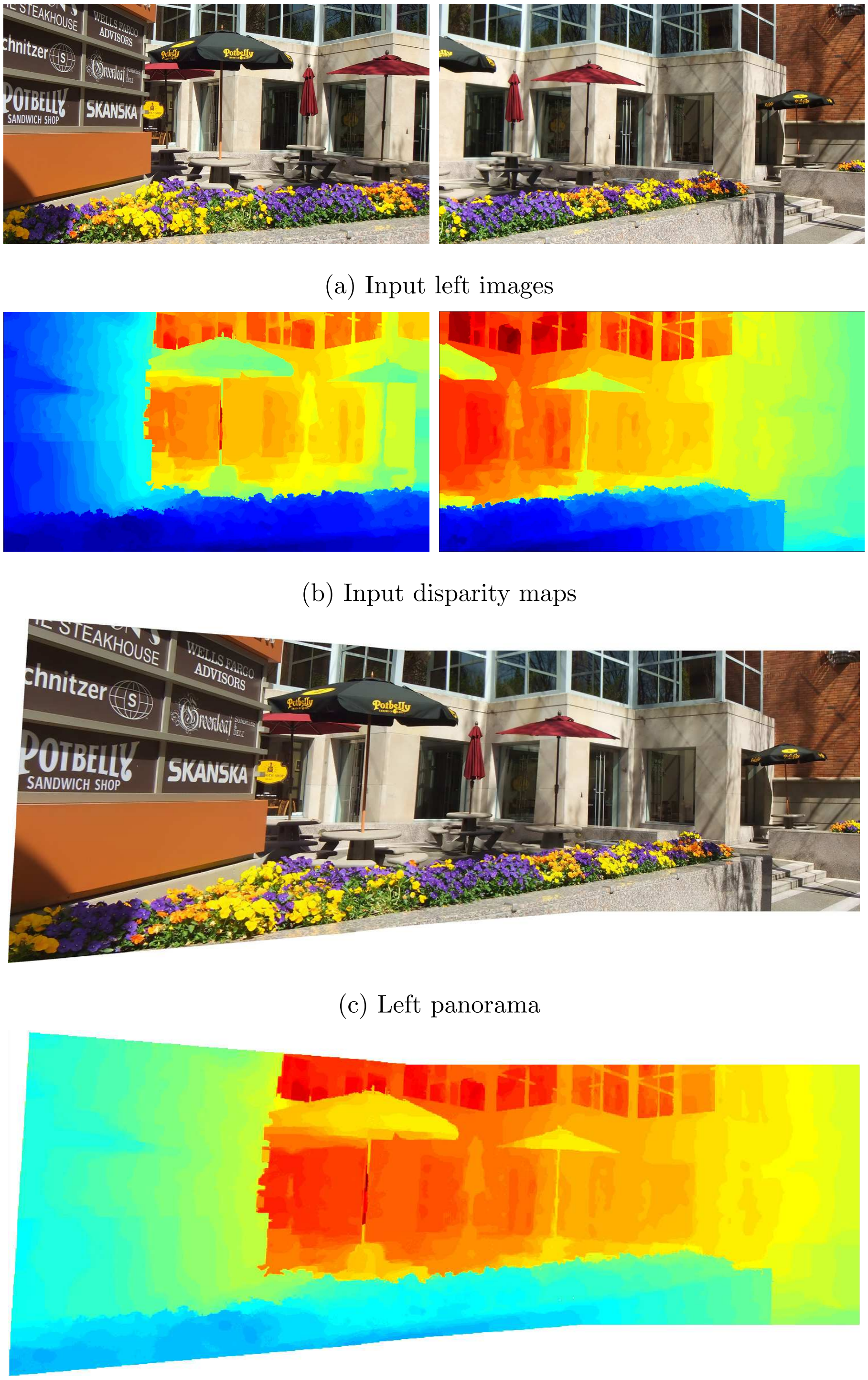
4.3.1 Left Panorama Stitching

Our method starts by stitching one of the two views of a stereoscopic panorama. Without loss of generality, our method selects to create the left panorama first. As our method allows a user to casually capture input stereoscopic images, there is parallax among the left input images. Actually, no matter how the input stereoscopic images are taken using a stereoscopic camera, parallax exists at least in one of the two views. Therefore, we choose to use monocular image stitching methods [17, 60, 61] that can handle parallax to create the left panorama.

Specifically, we use our parallax-tolerant monocular image stitching method [61]. This monocular stitching method first finds an optimal local alignment that allows for optimal stitching. The local alignment is a combination of homography-based warp and spatially-varying warp. Once input images are locally aligned, they are composed together using a seam-cutting algorithm [28] and a multi-band blending algorithm [8]. This step outputs the left panorama $\hat{I}_l^p$ as well as the intermediate stitching information that will be used in later steps, including the warped left images and the seam where the warped images are merged.

4.3.2 Target Panoramic Disparity Map Estimation

A stereoscopic image has an extra dimension of disparity, which controls the perceived depth [33]. To generate a good stereoscopic panorama, we need to not only stitch the input images, but also seamlessly stitch the disparity maps of input images to ensure proper 3D depth perception. We stitch the disparity maps in the disparity gradient domain using a Poisson blending method [39] and obtain the target disparity map $\hat{D}^p$ of the stereoscopic panorama. Specifically, we minimize the following energy function that aims to preserve the disparity gradients of the



(a) Input left images

(b) Input disparity maps

(c) Left panorama

(d) Target disparity map

Figure 4.2: Target panoramic disparity estimation.

input stereoscopic images I_1 and I_2.

$$\sum_{\hat{d}_i} \sum_{j \in N_i} \|(\hat{d}_i - \hat{d}_j) - dd_{i,j}\|^2, \tag{4.1}$$

$$\text{where } dd_{i,j} = \begin{cases} d_{1,i} - d_{1,j} & \text{if } l_i = 1 \\ d_{2,i} - d_{2,j} & \text{if } l_i = 2 \end{cases}$$

where $\hat{d}_i$ and $\hat{d}_j$ are the target disparities at neighboring pixels i and j of the left panorama $\hat{I}_l^p$ and N_i is the four-connected neighborhood of pixel i. $dd_{i,j}$ is the disparity difference between pixel i and j in the proper input stereoscopic image. If pixel i in the left panorama comes from the input image $I_{1,l}$, which is indicated by its label $l_i = 1$, $dd_{i,j}$ takes the disparity difference in the input stereoscopic image I_1. These labels come from the seam-cutting step in creating the left panorama. Similarly, if pixel i comes from the input image $I_{2,l}$, $dd_{i,j}$ takes the disparity difference in the input stereoscopic image I_2. Here the input disparity values like $d_{1,i}$ can be obtained by finding the corresponding pixel in the input image according to the warping applied to create the left panorama and taking the corresponding disparity value. Finally, we set the boundary condition of the above energy minimization problem by keeping the original disparities of the pixels that originally come from $I_{1,l}$ and are out of the overlapping region. Figure 4.2 shows an example of target panoramic disparity estimation.

Since a panoramic image typically contains a large number of pixels, the above Poisson's equation involves a large number of variables. To make this step efficient, we divide the left panorama into a uniform grid mesh and only compute the disparities for the grid vertices. Our experiments show that the mesh cell size of 5×5 pixels works well. This step outputs $\hat{D}^p$, the target disparity map of the final panorama (even before we create it). A user can further edit this target disparity map to manipulate the stereoscopic 3D viewing experience using tools like non-linear disparity mapping [29]. The disparity maps of all the results in this chapter are directly from the above Poisson blending algorithm and they are not post-edited unless otherwise noted.

4.3.3 Right Panorama Stitching

After we obtain the target disparity map $\hat{D}^p$ of the stereoscopic panorama, we first warp the right images $I_{1,r}$ and $I_{2,r}$ of the input stereo images according to the target disparity map. This warping step aligns the right input images as they are warped accordingly to the same target disparity map. Compared to the common method that aligns images based on the feature correspondence between these images, this approach has an important advantage in that the alignment result better respects the target disparity map and avoids introducing vertical disparities. Once we warp the right input images, we stitch these warped images using an extended seam-cutting method guided by the left panorama to create the right panorama.

Right input image warping

All the right input images are warped in the same way. For simplicity and clarity, we omit the subscripts $\{1, 2\}$ here. For each grid vertex in the left panorama, we first find its corresponding point in the corresponding left input image by inverting the warping used to align left images to create the left panorama. We then find its corresponding point in the right image according to the input disparity map. In this way, we obtain a set of control points in the right image, denoted as $\{\mathbf{p}_{r,i}\}$. Their corresponding points in the left panorama are $\{\hat{\mathbf{p}}^p_{l,i}\}$. The disparities of these control points are known from $\hat{D}^p$. Our method uses these control points to guide the warping of each right image.

Various spatially-varying warp methods have been developed to warp an image guided by a set of control points [27, 29, 31, 60, 61]. We extend these methods to warp each right input images guided by the set of control points. Specifically, we divide each right image into a uniform grid mesh and formulate image warping as a mesh warping problem, where the unknowns are the coordinates of mesh vertices. The mesh warping problem is defined as a quadratic minimization problem that enforces the disparities of the control points and minimizes visual distortion. We describe the energy terms below.

Disparity term. Our method encourages the control points to have the target disparities so that the stereoscopic panorama can deliver proper depth perception to viewers. Since each control point $\mathbf{p}_{r,i}$ in the right-view image is not necessarily a grid vertex, we first find the grid cell that encloses the control point in the right image and then represent it as a linear combination of the four vertices of the cell. The combination coefficients w_j are computed using the inverse bilinear interpolation method [20]. These coefficients are then used to combine the vertices $\hat{\mathbf{v}}_j$ in the output image to compute the location of the control point in the output image. We define the disparity energy term below.

$$E_d = \sum_{\mathbf{p}_{r,i}} \| \sum_j w_j \hat{\mathbf{v}}_j - \hat{\mathbf{p}}^p_{l,i} - \hat{\mathbf{d}}_i \|^2 \tag{4.2}$$

where $\hat{\mathbf{d}}_i$ is the target disparity vector of the control point $\mathbf{p}_{r,i}$, taking the form $\hat{\mathbf{d}}_i = [\hat{d}_i \ 0]^T$, where $\hat{d}_i$ is the target (horizontal) disparity and the vertical disparity is set 0. $\hat{\mathbf{p}}^p_{l,i}$ is the corresponding point of $\mathbf{p}_{r,i}$ in the left panorama.

Global alignment term. The disparity term only directly constrains warping of the image region with control points. These control points, however, only exist on one side of the stitching seam in the left image that is finally selected to make the left panorama, as illustrated in Figure 4.3. For the regions with no control points, warping often distorts them. To solve this problem, we first estimate the

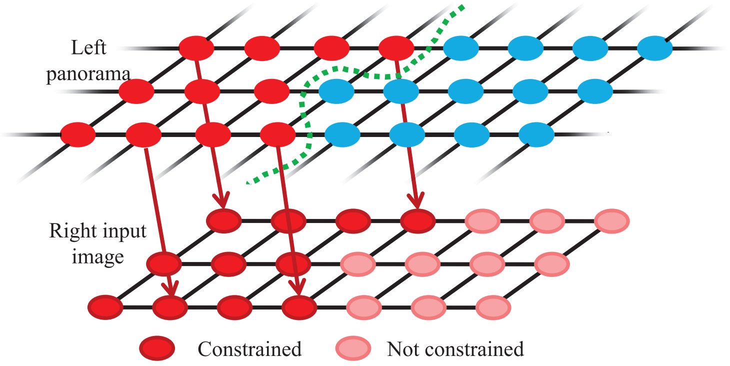


Figure 4.3: Control points. The control points only exist on one side of the stitching seam. Thus a part of the right-view image (with the light-red points) has no disparity constraint and will be distorted during warping if not taken care of.

best-fitting homography according to the control points and then employ this best-fitting homography to pre-warp the right input image. As the pre-warping result often provides a good approximation, our method encourages the regions without control points to be as close to the pre-warping result as possible. We define the global alignment term as follows

$$E_g = \sum_i \tau_i \|\hat{\mathbf{v}}_i - \bar{\mathbf{v}}_i\|^2 \tag{4.3}$$

where $\hat{\mathbf{v}}_i$ and $\bar{\mathbf{v}}_i$ are the corresponding vertex in the warping result and in the pre-warping result. τ_i is a binary value. We set it 0 if there is a control point in the neighborhood of $\hat{\mathbf{v}}_i$; otherwise it is 1.

Smoothness term. To minimize visual distortion, our method encourages each grid cell to undergo a similarity transformation. We use the quadratic energy term from [23] to encode the similarity transformation constraint.

$$E_s = \sum_{\hat{\mathbf{v}}_i} w_i \|\hat{\mathbf{v}}_i - (\hat{\mathbf{v}}_j + u(\hat{\mathbf{v}}_k - \hat{\mathbf{v}}_j) + v\mathbf{R}(\hat{\mathbf{v}}_k - \hat{\mathbf{v}}_j))\|^2 \tag{4.4}$$

where $\hat{\mathbf{v}}_i$, $\hat{\mathbf{v}}_j$, and $\hat{\mathbf{v}}_k$ are every three vertices of a grid cell in the output mesh. w_i is the average saliency value inside the triangle defined by the three vertices and

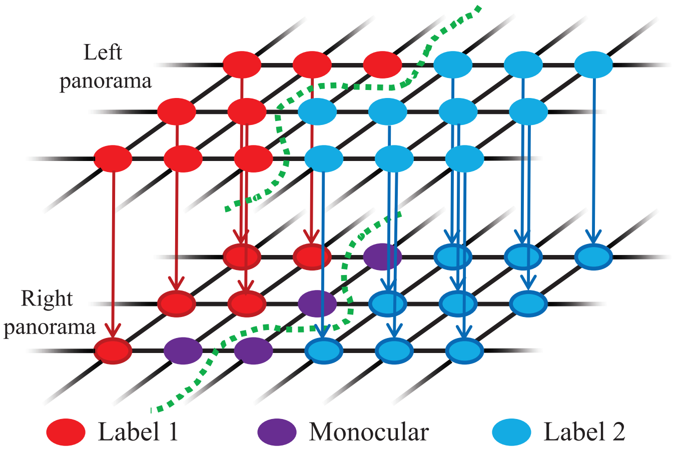


Figure 4.4: Seam-cutting for the right panorama. The labels, 1 or 2, from the left-view seam-finding result are propagated to the corresponding pixels in the right view (bottom). Pixels (in purple) in the monocular region in the right view will not have recommended labels from the left view. Propagated labels from the left panorama (top) are encoded as soft constraints in Equation 4.6 to tolerate left-right matching errors or make trade-off for the monocular stitching quality in Equation 4.7.

is computed using the same method as [31]. u and v are the coordinates of $\mathbf{v}_i$ in the local coordinate system defined by $\mathbf{v}_j$ and $\mathbf{v}_k$, where $\mathbf{v}_i$, $\mathbf{v}_j$, and $\mathbf{v}_k$ are the corresponding vertices in the input mesh of the right image. $\mathbf{R} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$.

We combine the above energy terms and obtain the following linear least squares problem.

$$E = E_d + \lambda E_g + \gamma E_s \tag{4.5}$$

where λ and γ are weights with default values 0.7 and 0.4, respectively. We solve this energy minimization problem using a sparse linear solver. The outputs from this step are the warped right images $\hat{I}_{1,r}$ and $\hat{I}_{2,r}$ according to the target disparity map of the stereoscopic panorama.

Seam-cutting for right panorama stitching

We develop a seam-cutting method to stitch the warped right images $\hat{I}_{1,r}$ and $\hat{I}_{2,r}$ guided by the seam-cutting result in creating the left panorama. The goal is to create the right panorama such that it is consistent with the left panorama. We extend the seam-cutting method for monocular image stitching [28] with an extra energy term to handle the stereo consistency problem. Specifically, we formulate this seam-cutting problem as a labeling problem. For each pixel in the overlapping region, we aim to assign it with a label either 1 or 2, indicating the pixel coming from $\hat{I}_{1,r}$ or $\hat{I}_{2,r}$. We now describe the energy terms for this labeling problem.

Stereo consistency term. Our method encourages pixels in the right panorama to take the same labels as their corresponding pixels in the left panorama, as shown in Figure 4.4. Therefore, for each pixel that a corresponding pixel in the left panorama can be found for, we encourage it to take the same label as the corresponding pixel in the left view.

$$E_{sc}(L) = \sum_{i \in \mathcal{S}} \rho_i \delta(l_i != l_i^l) \tag{4.6}$$

where $\mathcal{S}$ is the set of pixels in the right panorama that we can find corresponding pixels in the left panorama for.

L is the labeling map for pixels in the right panorama, l_i is the label for pixel i, and l_i^l is the label of the corresponding pixel in the left panorama. $\delta(l_i != l_i^l)$ is an indicator function that takes value 1 if $l_i != l_i^l$ and 0 otherwise. ρ_i is a weight that measures the confidence of matching pixel i between the left and right view, which is computed based on the color difference between pixels/patches or is available from the output of many optical flow and stereo matching algorithms.

Monocular color term. To create a seamless right panorama, our method aims to minimize the color difference between the overlapping regions of the $\hat{I}_{1,r}$ and $\hat{I}_{2,r}$

along the seam. Consider two adjacent pixels i and j in the overlapping region. If these two pixels take different labels, the color difference between $\hat{I}_{1,r}$ and $\hat{I}_{2,r}$ at pixel i and j should be as small as possible.

$$
\begin{aligned}
E_{mc}(L) = \sum_{i,j}(d(i, \hat{I}_{1,r}, \hat{I}_{2,r}) + d(j, \hat{I}_{1,r}, \hat{I}_{2,r}))\delta(l_i! = l_j) \\
d(i, \hat{I}_{1,r}, \hat{I}_{2,r}) = \|\hat{I}_{1,r}(i) - \hat{I}_{2,r}(i)\|^2
\end{aligned} \tag{4.7}
$$

where $d(i, \hat{I}_{1,r}, \hat{I}_{2,r})$ is the color difference at pixel i between $\hat{I}_{1,r}$ and $\hat{I}_{2,r}$, and $\delta(l_i! = l_j)$ is an indicator function, taking value 1 if $l_i! = l_j$ and 0 otherwise.

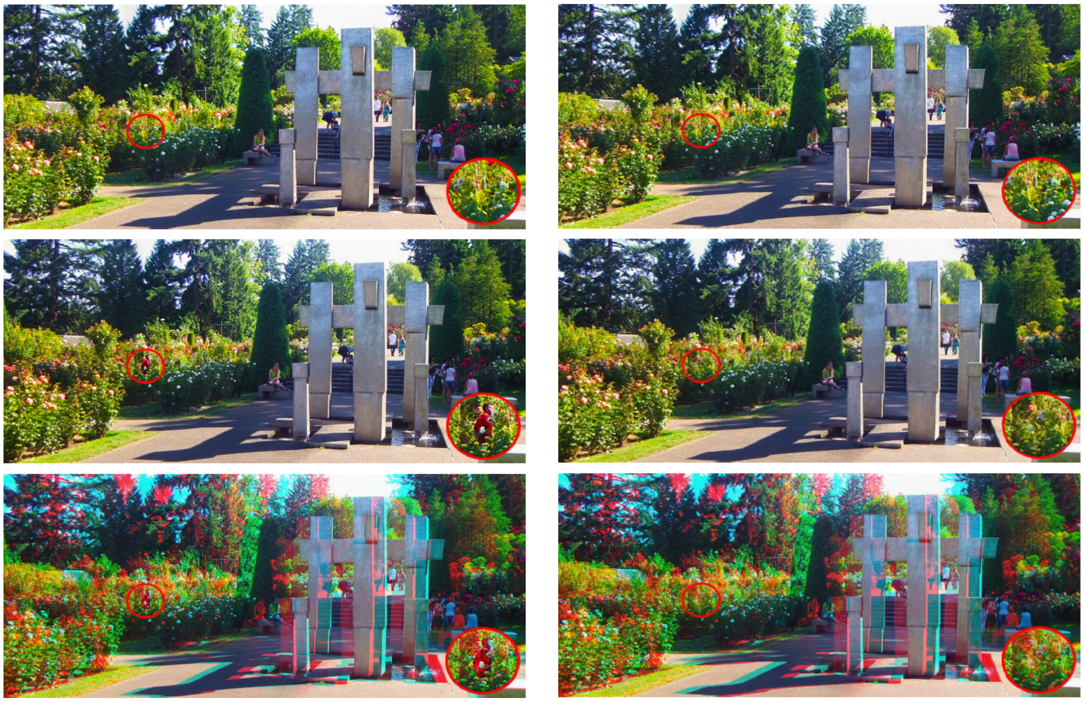
We combine the above terms and get the following minimization problem that aims to find an optimal labeling map.

$$
\begin{aligned}
E(L) = \alpha E_{sc}(L) + E_{mc}(L) \\
\text{s.t. } \; l_k = \begin{cases} 1 & \text{if } k \in \hat{I}^m_{1,r} \\ 2 & \text{if } k \in \hat{I}^m_{2,r} \end{cases}
\end{aligned} \tag{4.8}
$$

where α is a parameter with default value 0.5. $\hat{I}^m_{1,r}$ denotes the non-overlapping region in $\hat{I}_{1,r}$ where pixels take label 1. Similarly, pixels in the non-overlapping region $\hat{I}^m_{2,r}$ take label 2. We solve the above labeling problem using a standard graph-cut algorithm. After we find the seam, we use the seam and the multi-band blending algorithm [8] to compose the final right panorama.

4.4 EXPERIMENTS

We experimented with our stereoscopic image stitching methods on a variety of images taken by stereo cameras *Fujifilm* FinePix 3D W3 and *Panasonic* HDC-Z10000. These input stereo images were casually taken by these handheld cameras and therefore both the left images and right images exhibit large parallax. We compare our method to a baseline solution that employs a state-of-the-art monocular stitching method to stitch the left and right panorama independently [61].



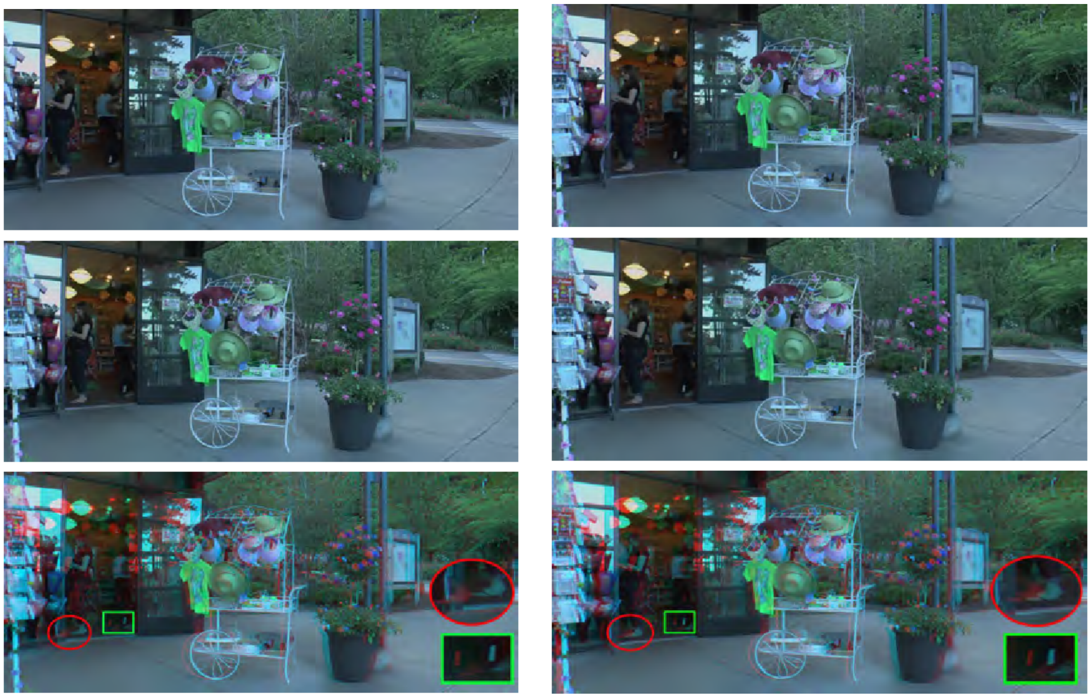
(a) Baseline result (Independent stitching)

(b) Our result

Figure 4.5: Comparison between independent stitching results and our results. In each column, we show the left-view, right-view and red-cyan anaglyph of the stereo panorama.

Since the baseline method creates the left and right panorama independently, the disparity distribution is often problematic. For the panoramas from the baseline method, we manually shifted the left and right panorama vertically so that there is as small vertical disparities as possible in the main object. We also shifted them horizontally so that the horizontal disparities in the main object are as similar to the corresponding panoramas created by our method as possible. Our results were not adjusted.

Figure 4.5 shows the left, right, and red-cyan anaglyph versions of the stereoscopic panoramas stitched by the baseline solution and our method. The baseline



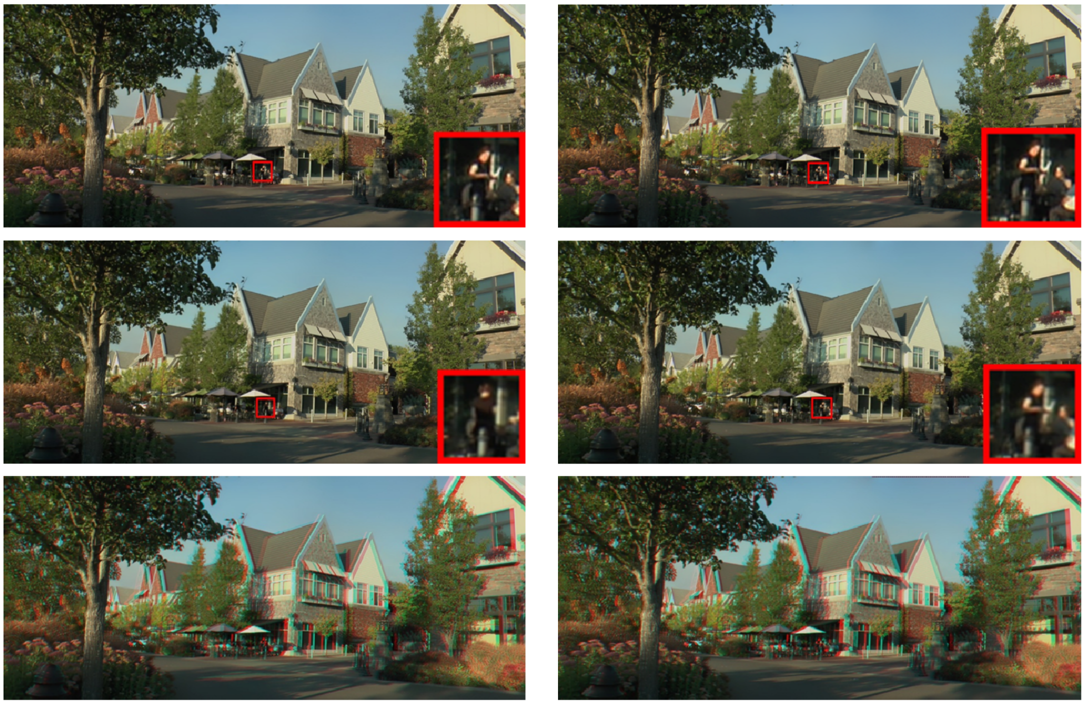
(a) Baseline result (Independent stitching)

(b) Our result

Figure 4.6: Comparison between independent stitching results and our results. In each column, we show the left-view, right-view and red-cyan anaglyph of the stereo panorama.

method independently creates the left and right panorama and thus cannot ensure the consistency between the two panoramas. For example, a person in red T-shirt appears in the left panorama but disappears in the right one, as shown in Figure 4.5(a). This inconsistency brings in the “monocular object violation” [33]. This is because different seams are used to stitch the left and right panorama. Our method stitches the right panorama constrained by the left panorama and is free from this monocular object violation, as shown in Figure 4.5(b).

Figure 4.6 shows another example. Although we manually aligned the left and right panorama of the baseline result, significant vertical disparities still exist, as shown in Figure 4.6(a), which will cause “3D fatigue” [33]. Since our method warps



(a) Baseline result (Independent stitching)

(b) Our result

Figure 4.7: Comparison between independent stitching results and our results. In each column, we show the left-view, right-view and red-cyan anaglyph of the stereo panorama.

the right input images according to the target panoramic disparity map, our result is free from the vertical disparity artifacts, as shown in Figure 4.6(b).

Figure 4.7 shows the third example. Since the baseline method cannot consistently stitch the left and right images, a monocular object has been introduced into the stereo stitching result. The waitress in the left and right stitching results show two different postures, as shown in the top two images in Figure 4.7(a). Such inconsistency will cause severe retinal rivalry and lead to visual fatigue. In contrast, our method can generate consistent left and right stitching result, as shown in Figure 4.7(b), which delivers comfortable viewing experience.

4.4.1 User Study

We conducted a user study to evaluate the user experience of viewing stereoscopic panoramas created by our method and the baseline method. Our study displayed stereoscopic panoramas on an ASUS VG236H 3D monitor with shuttered glasses. We selected 10 sets of input stereo images. For each set, we created two stereoscopic panoramas, one using our method and the other using the baseline solution. We obtained 20 stereoscopic panoramas in total.

There were 10 participants in our study, including 4 females and 6 males. They are students from various departments, including computer science, civil engineering, chemistry, biology, etc. They all have normal stereopsis perception. They do not know how each panorama was created. Before the study, we provided four panoramas, two from each method, for them to look at to get used to viewing stereoscopic panoramas. In our study, we showed the 20 stereoscopic panoramas mentioned above to each participant one by one in a random order. Participants can look at a panorama as long as they want. After a participant finishes looking at a panorama, we ask three questions.

1. Is it easy for you to perceive 3D?
2. Do you feel comfortable viewing the panorama?
3. Are you satisfied with the quality of the panorama?

The participant rated each question using a Likert scale ranging from 1 to 5, with 5 being the most positive. We report the average scores (μ) and the standard deviations (σ) in Table 4.1. Our study confirms our hypothesis that independently creating the left and right view of a stereoscopic panorama is problematic and will damage the stereoscopic 3D viewing experience. The average comfort rate for the baseline results is 2.68. In contrast, our results deliver a more comfortable 3D

	3D		Comfort		Quality	
	mean	std	mean	std	mean	std
Ours	4.10	0.52	4.19	0.61	3.84	0.45
Baseline	3.78	0.72	2.68	0.46	2.70	0.62

Table 4.1: User study results.

viewing experience with the average rate 4.19. Similarly, users are more satisfied with our results (μ=3.84, σ=0.45) than the baseline results (μ=2.70, σ=0.62). The p-values of the paired two sample t-test between our results and the baseline results for both comparisons are smaller than 0.001, which shows that the difference between the two sets of results are very significant.

The stereoscopic panoramas from both our method and the baseline method can allow users to easily obtain 3D perception and our results (μ=4.10, σ=0.52) are easier for users than the baseline result (μ=3.78, σ=0.72). The p-value for the study is 0.123, which shows that the difference between the two sets of results is not statistically significant.

The post-study informal feedback shows that participants complained most that they cannot fuse the left and right view to obtain 3D perception for some regions in some panoramas. We found that this problem is due to the inconsistency between the left and right panorama. For example, a visually salient object only appears in one of the two views, which brings in “retinal rivalry”. The inconsistency problem only occurs in the baseline results.

4.4.2 Discussion

Compared to the monocular stitching method [61], our three-step approach adds two extra steps besides a standard pre-processing step to estimate disparity maps of input images: target panoramic disparity map stitching and right panorama stitching. Our method solves a Poisson's equation to stitch the target disparity map. On two input images with size about 900 × 600, this step takes less than 1 second using a desktop machine with *Intel* i7 CPU and 16 GB memory. Right panorama stitching has two main computational steps: spatially-varying warping and seam-cutting using graph-cut. These two take less than 1 second in total.

Our method relies on the disparity maps of input stereo images to take care of disparity and consistency issues during stitching. The input disparity maps sometimes contain errors; however, we found that our method is robust against the disparity errors. For example, we use input disparities to establish the left-right correspondences and propagate the labels from the left panorama to the right one. When a pixel in the right panorama is mapped onto a wrong pixel in the left panorama, the label recommended for it will mostly be corrected still as neighboring pixels in the left panorama often share the same label except across the stitching seam. Moreover, our method encodes labeling propagation as a soft constraint. The errors can usually be corrected by the other energy term in the optimization. Similarly, while the warping of the right images is guided by the target disparity map, disparity errors in a few pixels will be corrected by the smoothness term of the warping energy function.

The first step of our method uses our parallax-tolerant image stitching technique to stitch the left panorama [61]. While this monocular stitching method can handle parallax well in general, when the parallax among images from the same view is very large, it sometimes suffers from alignment artifacts. As our method builds upon this first step to produce the right panorama, the right panorama produced by our method also shares similar artifacts. Since our three-step approach is flexible, it can easily replace the monocular stitching method that is currently used with a more advanced monocular stitching method in future.

Chapter 5

VIDEO STITCHING

The previous chapters explore both regular 2D and stereoscopic 3D image stitching techniques. In this chapter, we discuss an important extension of image stitching: video stitching. We present two techniques to tackle the video stitching problem.

5.1 INTRODUCTION

SLR cameras, smart phones, action cameras and video surveillance systems all provide video recording functions. Users can conveniently take multiple videos using a multi-camera bracket mount. Stitching them together to create wide field of view videos would not only meet the entertainment needs of general users, such as virtual reality (VR) exploration, but it also would better assist scene analysis and explorations, such as suspicious activity detection.

This dissertation intends to develop a stitching technique that can stitch multiple pre-synchronized videos. The input videos can be captured from a camera array which contains multiple cameras with fixed inter-camera configurations. Different lenses and camera models are allowed, and the camera array can be either fixed at a tripod or hand-held by users. Figure 5.1 shows the multi-camera capture system where two cameras are mounted using a dual camera bracket; one camera is Nikon D800, the other one is Sony NEX3N. A great degree of freedom was given in building the multi-camera capture system to make the video stitching technique applicable to amateur users.

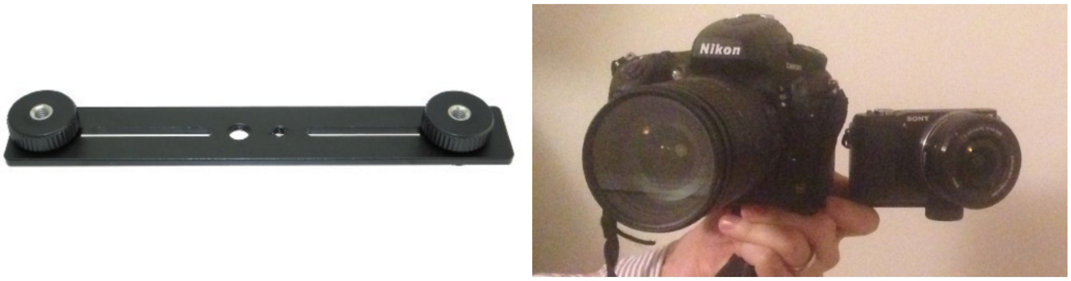


Figure 5.1: Dual camera capture system.

Different from stereoscopic images which have an extra dimension of depth, videos have an extra dimension of time. Similar to stereoscopic images, treating video stitching as a batch image stitching task to stitch every video frame independently and combine them together to create video panoramas is problematic as different frame stitching results may lead to incoherent frame transitions. Several alternatives can be used to stitch videos with better quality. For example, panoramic videos can be stitched by consistently picking pixels from specific cameras in the overlapping region. In this way, temporal coherence can be well maintained in the panoramic video results. Nevertheless, the panoramic videos will suffer from severe broken structure artifacts in the overlapping region since no alignment has been involved in the stitching process. Another alternative is to estimate a reference alignment and use the same alignment to stitch every frame. However, for videos with significant movements, the same alignment could not work for all frames. Misalignments can be easily introduced into the results. Therefore, stitching results created using this method often suffer from ghosting artifacts, broken structure artifacts or temporal incoherent frame transitions. Figure 5.2 shows one frame of a panoramic video demo created using VideoStitch Studio [51]. We can see clear ghosting artifacts shown in the red rectangle. Temporal incoherence and frame inconsistency could largely compromise the viewing

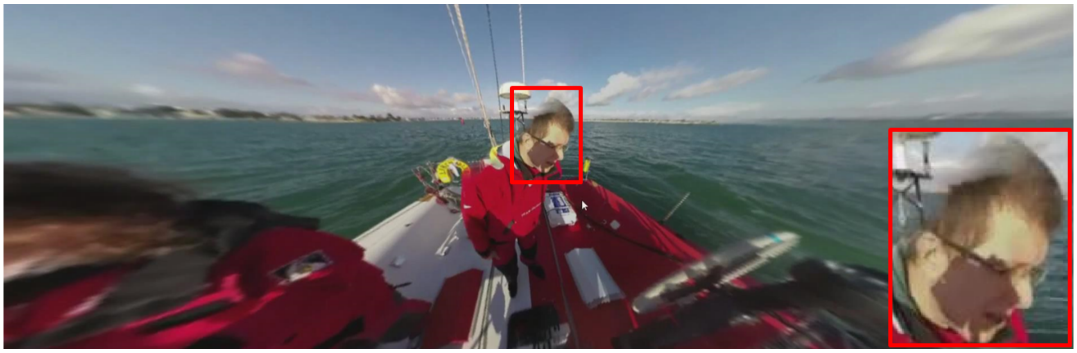


Figure 5.2: Video panorama created by VideoStitch Studio.

experience, especially for 3D content such as virtual reality videos. Thus, maintaining temporal coherence and frame consistency is crucial for delivering a high quality video stitching result. Google Jump and Facebook 360 both provide 360 panoramic stereoscopic video capture solutions; however, their stitching algorithms rely on specific camera setups which are not accessible to amateur users. Most importantly, their solutions require that the camera array must be fixed on either a tripod or a moving rig with constant velocity in a straight line. If users violate such shooting guidelines, artifacts could be introduced into the final results.

5.2 RELATED WORK

Extensive research has been done in the image stitching area, but not enough work has focused on stitching media content in the video domain. Traditional techniques use panoramic video capture systems [10, 19] to create video panoramas. However, these systems rely on specific camera arrays and cannot handle parallax well, and thus they can only work properly when dealing with scenes that do not have close objects in the overlapping area. New approaches such as OMNICAM [46], GoPano [18] and FullView [15] can handle parallax in a better way, but these approaches still rely on specific mirror rigs [34, 35]. All those solutions are limited by their flexibilities and are not accessible by amateur users.

Rav-Acha *et al.* [42] created dynamosaics, in which events that happened at different times all play simultaneously, using videos captured by a moving camera scanning a dynamic scene. Agarwala *et al.* [4] developed a method to create panoramic video texture which has an infinitely playing panoramic video with periodic and localized motion. Tompkin *et al.* [56] and Pirk *et al.* [40] embedded video clips into panoramic contexts to allow users to explore panorama scenes better. These methods have been proven to work reasonably well on videos that contain objects with localized and periodic movements, but they cannot handle videos with significantly moving objects.

Zhi *et al.* [62] presented a Depth-Based Dynamic Mosaic (DMB) approach which performs foreground-background segmentation first, and then it projects foreground dynamic objects onto the stitched background panorama plane according to their depth estimation for temporal coherence maintenance. This method can effectively handle videos with a single moving object but cannot work well for videos with complex scenes and motion patterns. Perazzi *et al.* [38] presented an algorithm that uses weighted extrapolation of warps in non-overlapping regions to ensure temporal coherence; the algorithm also relieves the global deformation using constrained relaxation. Jiang *et al.* [25] proposed using a spatial-temporal local warping method to maintain temporal coherence.

5.3 MOTION MAP GUIDED VIDEO STITCHING

In this section, we present a dense motion map guided video stitching technique. We take a set of pre-synchronized videos captured from a fixed or hand-held camera array as input. The camera array contains multiple cameras with fixed inter-camera configurations. Our output is a stitched wide field of view panoramic

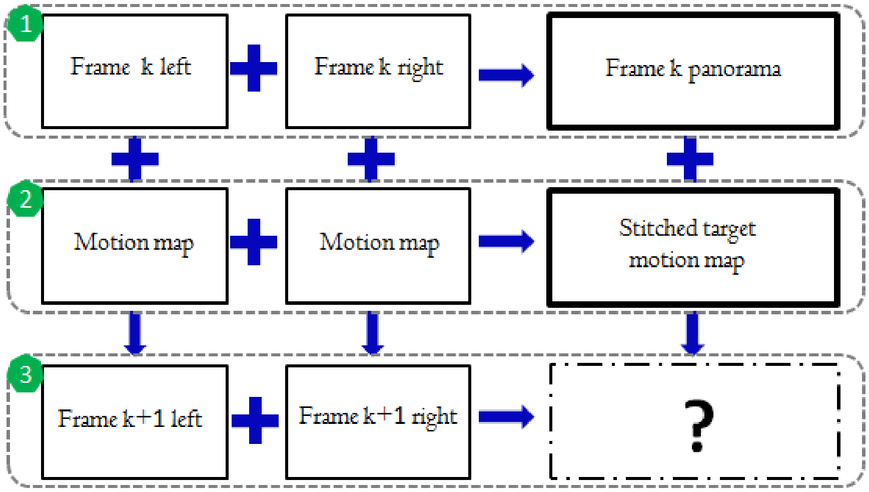


Figure 5.3: Naive video stitching pipeline.

video. One naive way to stitch videos is to stitch every frame separately and then combine them together. Since each frame stitching may use completely different transformations, it is impossible to maintain temporal coherence with this method. A better solution is to estimate transformations among the first frames and then use the same transformations for every other frame. This method could work on occasion if the visual scene is static or the scene only contains very few moving objects of a small size. However, video content may change largely frame by frame if the shooting camera is moving or salient objects in the scene are moving. As a result, fixed transformations cannot handle different frame content alignments.

Inspired by our stereoscopic stitching method, we propose to maintain the temporal coherence of the original input videos by preserving the motion field of the input videos. As shown in Figure 5.3, a naive 3-step video stitching method can be summarized as follows.

Start from the k^{th} frame and with k=1 by default,

1. Stitch the current frame of input videos using a parallax-tolerant image

stitching technique to obtain the current frame panorama and use it as the reference panorama.

2. Estimate the motion field between the current frame and its successive frame for each video. Then use the same warping function that was applied to the current frame to warp the motion fields. After that, stitch the warped motion field together seamlessly with the guidance of the reference panorama seam finding result to create the target panoramic motion map.

3. Warp and stitch the successive frame of each input video to create a successive panorama according to the target panoramic motion map and the reference panorama. Update the reference panorama with the new generated successive panorama. Then repeat step 2 and 3 to stitch the rest of frames.

However, this method can only maintain temporal coherence in theory: it needs to address several challenges to achieve temporal coherence in practice. First, this method needs to handle accumulated motion estimation errors. In particular, the left frame and the right frame can be aligned very well with parallax-tolerant image stitching at the first frame. However, when using the target panoramic motion map to guide the successive frame warping, motion estimation errors of the input videos are often introduced into the results. Such motion errors could be accumulated across frames and eventually cause noticeable misalignment in the stitching result. Second, since users are allowed to freely move their shooting camera array, the cameras in the array actually cannot maintain perfect relative position. An extreme example is that one camera in the array remains fixed, and the other one rotates with the fixed camera as the rotating axis. Under this condition, the motion fields of the two videos are not consistent with each other. Two inconsistent motion fields will lead to different warping strength for two videos; in the end, the video with a greater warping strength would have more distortion and it cannot be matched with the other one.

To solve these problems, we improve our naive video stitching method by introducing four types of frames: independent frame (I), full-reference frame (F), reduced-reference frame (R), and semi-independent frame (S). Different stitching types can be selected according to frame types rather than using the same way for all frames. The particular features of each are described below:

- **I-frames** conduct frame warping independently and do not rely on any other frame warping results or motion fields. The stitching result created by independent frames is the *leading panorama*, which sets up the fundamental alignment model for the video stitching. One video stitching task only has one leading panorama.

- **F-frames** warp frames based on the previous frame warping result, the original motion fields and the target panoramic motion map between current frame and previous frame.

- **R-frames** do frame warping based on the original motion fields and target panoramic motion map. In a video, most of the frames are reduced-reference frames.

- **S-frames** do frame warping only based on target panoramic motion map. Since they rely on very limited reference information, they do the warping almost independently. At the same time however, they still take the previous frame warping result into consideration to maintain temporal coherence. The stitching result created by semi-independent frames are *semi-leading panoramas*.

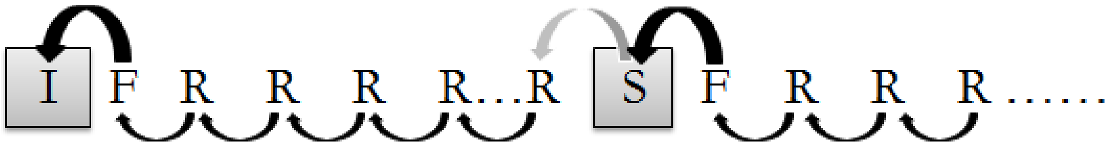


Figure 5.4: Video stitching frame dependencies.

The frame dependencies of the video stitching task is shown in Figure 5.4. In this way, motion estimation errors and the poor effects of inconsistent warping strength can be reset to zero every few frames, and thus cannot lead to any noticeable artifacts. The method is described in detail below. For simplicity, we consider the task of stitching two input videos V_l and V_r. More videos can be stitched similarly. Each input video has several frames. For example, f_l^1, f_l^2,..., f_l^n, are frame 1,2,..., n of V_l, respectively. We denote each frame stitching result of the output video panorama as O^1, O^2,..., O^n.

Our method pre-processes input videos to estimate the motion field between every two consecutive frames. We downsample each input video, estimate the motion field using the optical flow method [52], and scale up the resulting optical flow vectors to get the motion maps. To remove the perspective distortion, all our video frames are projected to a cylindrical plane before the stitching is performed.

5.3.1 Independent Frame Stitching

Our method starts with stitching f_l^k and f_r^k of V_l and V_r, by default $k = 1$. The first frame of input videos would always be stitched as I-frames. As the shooting camera array has multiple cameras with a certain baseline, parallax has been introduced into video frames. Therefore, we use our parallax-tolerant image stitching technique which can handle the parallax problem well enough to create first frame stitching result O^k as the *leading panorama*, as well as the intermediate stitching information that is used in later steps, including the warped frame $\hat{f_l^k}$ and $\hat{f}_r^k$ of V_l and V_r, and the seam where the warped frames $\hat{f}_l^k$ and $\hat{f}_r^k$ are merged.

5.3.2 Target Motion Map Estimation

Videos have an extra dimension of time. To make high-quality video panoramas, it is necessary to ensure the temporal coherence is maintained among all frames in the stitched video. Given two input motion maps M_l^k and M_r^k, we achieve this goal by stitching the motion maps in the motion gradient domain using a Poisson blending method [39] to obtain the target motion map M_O^k for frame f_l^k and f_r^k. Specifically, we minimize the following energy function that aims to preserve motion field of the input videos.

$$\sum_{\hat{m}_i} \sum_{j \in N_i} \|(\hat{m}_i - \hat{m}_j) - dm_{i,j}\|^2, \tag{5.1}$$

$$\text{where } dm_{i,j} = \begin{cases} m_{l,i} - m_{l,j} & \text{if } l_i = l \\ m_{r,i} - m_{r,j} & \text{if } l_i = r \end{cases}$$

where $\hat{m}_i$ and $\hat{m}_j$ are the target motion vectors at neighboring pixels i and j of O^k, and N_i is the four-connected neighborhood of pixel i. $dm_{i,j}$ is the motion difference between pixel i and j in the proper input video. If pixel i in O^k comes from f_l^k, which is indicated by its label $l_i = l$, $dm_{i,j}$ takes the motion difference between frame f_l^k and f_l^{k+1}. These labels come from the seam-cutting step in creating the leading panorama. Similarly, if pixel i comes from f_r^k, $dm_{i,j}$ takes the motion difference estimated between frame f_r^k and f_r^{k+1}. Here the input motion vectors such as $m_{l,i}$ can be obtained by finding the corresponding pixel in f_l^k according to the warping function applied to create the leading panorama and taking the corresponding motion vector. Finally, we set the boundary condition of the above energy minimization problem by keeping the original motion vectors of the pixels that originally come from f_l^k and are out of the overlapping region.

5.3.3 Full-reference Frame Stitching

In this step, we first estimate the warp guidance for f_l^{k+1} and f_r^{k+1} (successive frames of f_l^k and f_r^k) according to the leading panorama and motion maps. Then we warp f_l^{k+1} and f_r^{k+1} using content-preserving warping. Once we get the warped successive frames $\hat{f}_l^{k+1}$ and $\hat{f}_r^{k+1}$, we stitch them together using an extended seam-cutting method guided by the previous seam finding result to create the successive stitching result O^{k+1}.

Full-reference frame warping

Frames f_l^{k+1} and f_r^{k+1} are warped in the same way. For simplicity and clarity, we omit the subscripts $\{l, r\}$ here. As shown in Figure 5.5, (1) we first set a number of control points in the previous panorama O^k. For each control point $\{\mathbf{p}_O^{k,i}\}$, (2) we find its corresponding point in f^k by inverting the warping used for aligning f^k to create the previous panorama O^k. We denote them as $\{\mathbf{p}^{k,i}\}$. After that, (3) we find its corresponding point in f^{k+1} according to the input motion map M^k. In this way, we obtain a set of control points in f^{k+1}, denoted as $\{\mathbf{p}^{k+1,i}\}$. (4) We also find their corresponding points in the target panorama $\{\mathbf{p}_O^{k+1,i}\}$, which are computed by adding the motion vectors of these control points which are known from M_O^k to $\{\mathbf{p}_O^{k,i}\}$. Finally, (5) our method uses these control points to guide the warping of f_l^{k+1} and f_r^{k+1}.

Specifically, we divide f_l^{k+1} and f_r^{k+1} into a uniform grid mesh and formulate frame warping as a mesh warping problem, where the unknowns are the coordinates of mesh vertices. The mesh warping problem is defined as a quadratic minimization problem that enforces the motion of the control points and minimizes visual

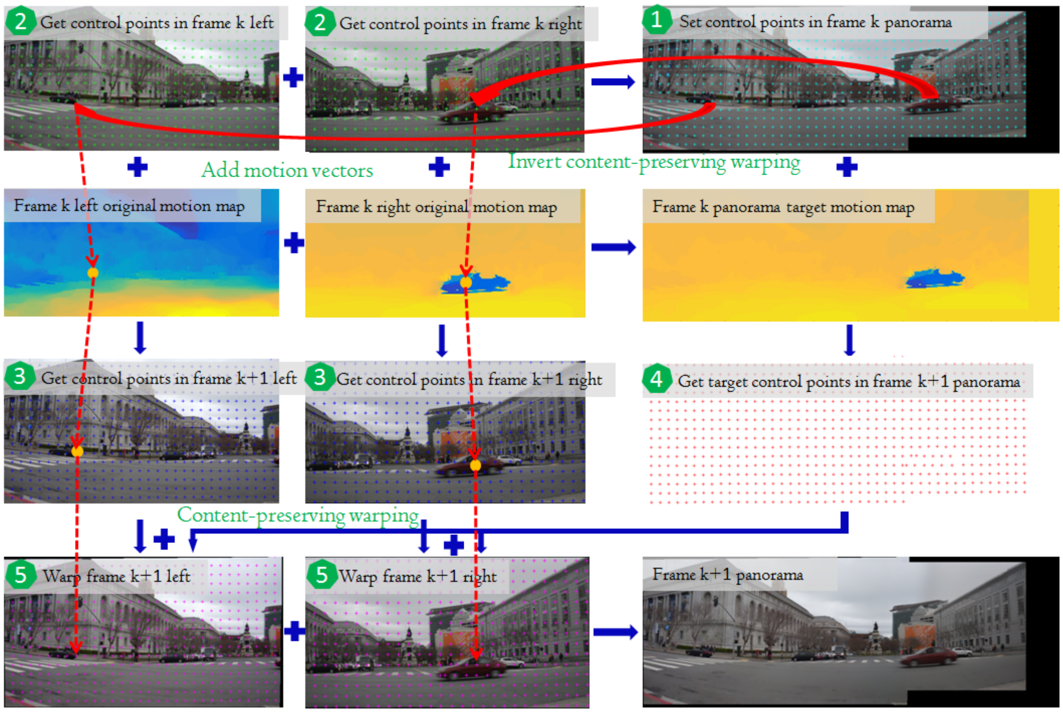

2 Get control points in frame k left
2 Get control points in frame k right
1 Set control points in frame k panorama
Add motion vectors
Invert content-preserving warping
Frame k left original motion map
Frame k right original motion map
Frame k panorama target motion map
3 Get control points in frame k+1 left
3 Get control points in frame k+1 right
4 Get target control points in frame k+1 panorama
Content-preserving warping
5 Warp frame k+1 left
5 Warp frame k+1 right
Frame k+1 panorama

Figure 5.5: Full-reference frame warping (green hexagons represent the step number). (1) Setup control points in stitched panorama; (2) obtain corresponding control points for original inputs; (3) and (4) obtain corresponding control points for successive original inputs and their stitched panorama by adding original motion vectors and target motion vectors; (5) warp successive original inputs according to control point positions and stitch them to get the successive panorama.

distortion. We describe the energy terms below.

Motion term. Our method encourages the control points to move according to the target motion vectors so that the temporal coherence of the input videos can be preserved. Since each control point $\{\mathbf{p}^{k+1,i}\}$ in f^{k+1} is not necessarily a grid vertex, we first find the grid cell that encloses the control point in f^{k+1} and represent it as a linear combination of the four vertices of the cell. The combination coefficients w_j are computed using the inverse bilinear interpolation method [20].

These coefficients are then used to combine the vertices $\hat{\mathbf{v}}_j$ in the output frame to compute the location of the control point in the output frame. We define the motion energy term as follows:

$$E_m = \sum_{\mathbf{p}^{k+1,i}} \| \sum_{j} w_j \hat{\mathbf{v}}_j - \mathbf{p}_O^{k,i} - \hat{\mathbf{m}}_i \|^2 \tag{5.2}$$

where $\hat{\mathbf{m}}_i$ is the target motion vector of the control point $\mathbf{p}^{k+1,i}$, taking the form $\hat{\mathbf{m}}_i = [\hat{hm}_i \ \hat{vm}_i]^T$, where $\hat{hm}_i$ is the horizontal motion vector and $\hat{vm}_i$ is the vertical motion vector. $\{\mathbf{p}_O^{k,i}\}$ is the corresponding point of $\mathbf{p}^{k+1,i}$ in the reference panorama O^k.

Smoothness term. To minimize visual distortion, our method encourages each grid cell to undergo a similarity transformation. We use the quadratic energy term from [23] to encode the similarity transformation constraint.

$$E_s = \sum_{\hat{\mathbf{v}}_i} w_i \| \hat{\mathbf{v}}_i - (\hat{\mathbf{v}}_j + u(\hat{\mathbf{v}}_k - \hat{\mathbf{v}}_j) + v\mathbf{R}(\hat{\mathbf{v}}_k - \hat{\mathbf{v}}_j)) \|^2 \tag{5.3}$$

where $\hat{\mathbf{v}}_i$, $\hat{\mathbf{v}}_j$, and $\hat{\mathbf{v}}_k$ are every three vertices of a grid cell in the output mesh. w_i is the average saliency value inside the triangle defined by the three vertices and is computed using the same method as [31]. u and v are the coordinates of $\mathbf{v}_i$ in the local coordinate system defined by $\mathbf{v}_j$ and $\mathbf{v}_k$, where $\mathbf{v}_i$, $\mathbf{v}_j$, and $\mathbf{v}_k$ are the corresponding vertices in the input mesh of the right image. $\mathbf{R} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$.

We combine the above energy terms to obtain the following linear least squares problem.

$$E = \lambda E_m + \gamma E_s \tag{5.4}$$

where λ and γ are weights with default values 1 and 0.3 respectively. We solve this energy minimization problem using a sparse linear solver. The outputs from this step are the warped successive frame $\hat{f}_l^{k+1}$ and $\hat{f}_r^{k+1}$ according to the target panoramic motion map of the reference panorama O_k.

Motion vector screening

We use the optical flow method [52] to estimate motion field. When pixel displacements between frames are very large, the optical flow method usually fails to give an accurate motion vector estimation. Motion estimation errors may also occur in textureless regions. To compensate motion estimation errors, we compute a motion confidence map and use it to eliminate control points with low confident motion vectors. Given two frames f^1, f^2 and the estimated motion map m^1 between them, for each pixel p^i in f^1, we use m^1 to compute its estimated corresponding pixel position $\hat{p^i}$ in f^2. After this, we calculate the color difference between these two corresponding pixels. The larger the color difference is, the more inaccurate the estimated motion vector is. Apart from this, occlusion also happens in video frames. Due to occlusion, some pixels in f^1 may not have corresponding pixels in f^2. For this reason, we estimate an occlusion map using the method described in [5]. Specifically, if two or more pixels in f^1 are mapped to the same pixel in f^2, the pixel with the largest motion vector value is the occluder while the rest are occluded. The computation produces an binary occlusion map O. If O(i, j) = 1, pixel (i, j) is an occluded, otherwise occluding pixel.

Temporal coherent seam finding

The same extended seam-cutting method as described in Chapter 4 is used here for stitching the warped successive frame $\hat{f}_l^{k+1}$ and $\hat{f}_r^{k+1}$ with the guidance of the seam-cutting result in creating O^k. The goal is to create O^{k+1} such that it is consistent with O^k. Specifically, we formulate this seam-cutting problem as a labeling problem. For each pixel in the overlapping region, we encourage pixels in the successive panorama O^{k+1} to take the same labels as their corresponding pixels in the reference panorama O^k. To create a seamless successive stitching result, we

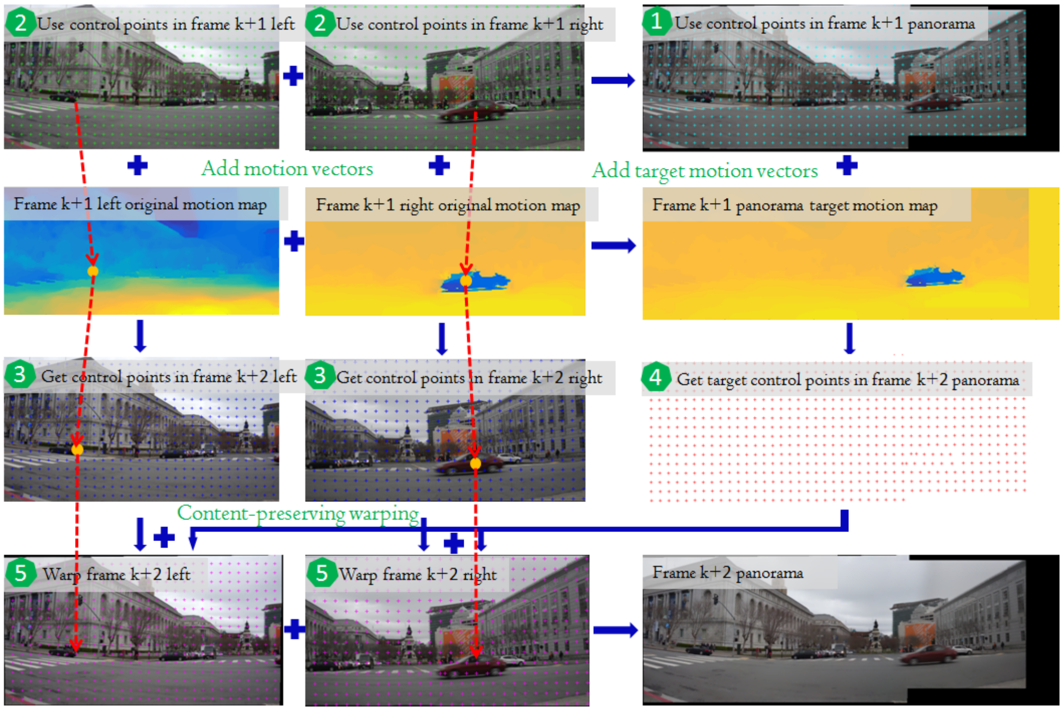

2 Use control points in frame k+1 left
2 Use control points in frame k+1 right
1 Use control points in frame k+1 panorama
Add motion vectors
Add target motion vectors
Frame k+1 left original motion map
Frame k+1 right original motion map
Frame k+1 panorama target motion map
3 Get control points in frame k+2 left
3 Get control points in frame k+2 right
4 Get target control points in frame k+2 panorama
Content-preserving warping
5 Warp frame k+2 left
5 Warp frame k+2 right
Frame k+2 panorama

Figure 5.6: Reduced-reference frame warping. (1) and (2) Use the same control points obtained from full-reference frame warping step as the control points for original inputs and stitched panorama; (3) and (4) obtain corresponding control points for successive original inputs and their stitched panorama by adding original motion vectors and target motion vectors; (5) warp successive original inputs according to control points and stitch them to get the panorama.

minimize the color difference between the overlapping regions of the $\hat{f}_l^{k+1}$ and $\hat{f}_r^{k+1}$ along the seam. We solve the above labeling problem using a standard graph-cut algorithm. After the seam is located, we use the seam and the multi-band blending algorithm [8] to compose the final $k+1$ panorama. We omit the mathematical equations here for the sake of space (refer to Chapter 4 for a detailed explanation on the equations).

5.3.4 Reduced-reference Frame Stitching

After obtaining the newly generated stitching result O^{k+1}, instead of finding a new set of control points by inverting the last frame warping results, we use the same set of control points that warped f_l^{k+1} and f_r^{k+1} to warp f_l^{k+2} and f_r^{k+2}. As shown in Figure 5.6, for each control point $\{\mathbf{p}^{k+1,i}\}$ in f_l^{k+1} and f_r^{k+1}, (1) we use the input motion map M^{k+1} to get their positions in f_l^{k+2} and f_r^{k+2}. For their corresponding points $\{\mathbf{p}_O^{k+1,i}\}$ in O^{k+1}, (2) we use the target motion map M_O^{k+1} to get their positions in O^{k+2}. Then (3) we use the same content-preserving warping technique described in the last section to warp f_l^{k+2} and f_r^{k+2}, and use the temporal coherent seam finding method to stitch the warped frames together to create the successive stitching results.

5.3.5 Semi-independent Frame Stitching

Simply relying on only one leading panorama and motion estimation to perform successive frame warping could have problems, because motion estimation errors can be accumulated so that tiny misalignments caused by motion estimation errors can eventually become large misalignments. To address this problem, a simple solution is to update the leading panorama every few frames using independent frame stitching. However, this method completely ignores the previous stitching result and could lead to temporal incoherence. Therefore, we use semi-independent frame stitching to generate *semi-leading panoramas* every few frames. The semi-independent frame interval number is experimentally set to 30 for videos with slow camera movement and 10 for videos with fast camera movement. A good semi-leading panorama should 1) preserve the geometric structures of the previous frames as much as possible to ensure a smooth transition between the previous

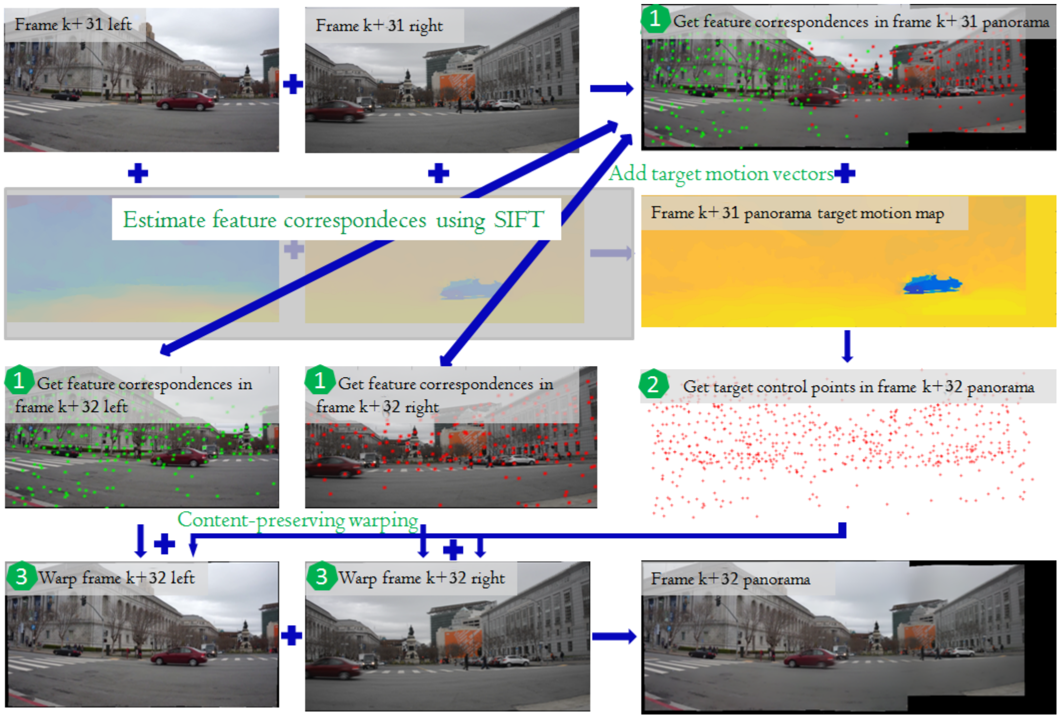
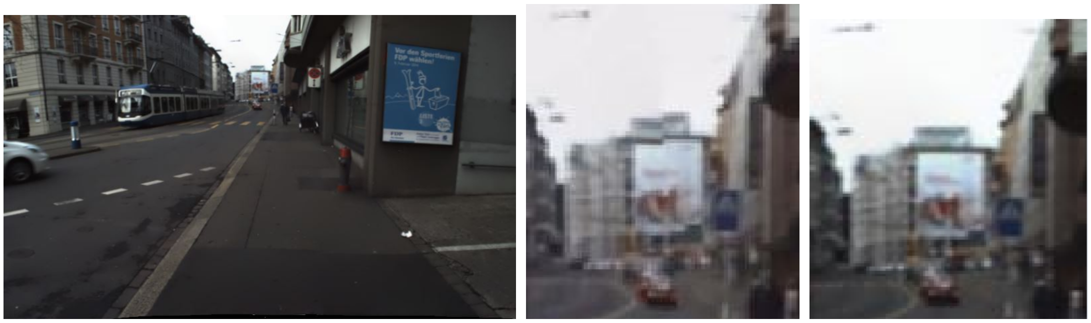

Frame k+31 left
Frame k+31 right
1 Get feature correspondences in frame k+31 panorama
Estimate feature correspondeces using SIFT
Add target motion vectors
Frame k+31 panorama target motion map
1 Get feature correspondences in frame k+32 left
1 Get feature correspondences in frame k+32 right
2 Get target control points in frame k+32 panorama
Content-preserving warping
3 Warp frame k+32 left
3 Warp frame k+32 right
Frame k+32 panorama

Figure 5.7: Semi-independent frame warping (green hexagons show the step number). (1) Estimate feature correspondences between current frame panorama and successive original input frames as the control points; (2) obtain target control points for successive stitched panorama by adding target motion vectors; (3) warp successive original inputs according to control points position and stitch them to get the successive panorama.

reduced-reference frame stitching result and the semi-leading panorama; 2) eliminate the matching errors that caused by accumulated motion estimation errors. To achieve these goals, different from independent frames which generate the leading panorama without using any reference information, semi-independent frames take the previous stitching results into consideration to maintain the temporal coherence. To decrease the poor effects of accumulated errors as much as possible, we



(a) (b) (c)

Figure 5.8: Comparisons between two video stitching methods. From left to right, the close-ups show comparisons to the methods: (b) Autopano Pro, (c) our result. Autopano Pro result has misalignment around the advertisement sign area. In contrast, our result does not introduce misalignment.

stitch the semi-independent frames using as little reference information as possible. Specifically, as shown in Figure 5.7, we (1) estimate feature correspondences between O^{k+31} and f_l^{k+32}, O^{k+31} and f_r^{k+32} using SIFT, and use those feature correspondences as the control points to warp f_l^{k+32} and f_r^{k+32}. With the target motion map M^{k+31}, (2) we can get the target positions of the control points in O^{k+32}. In this way, the misalignments caused by accumulated motion estimation errors can be reset to zero, at the same time, (3) content-preserving warping warps frame f_l^{k+32} and f_r^{k+32} with the guidance of the control points to maintain temporal coherence.

5.3.6 Experiments

In this section, we first introduce our multi-camera capture system. We then experimented with our method on a range of challenging videos with parallax and moving objects. We also compared our method to the state-of-the-art methods, including VideoStitch Studio [51] and Autopano Pro [26], which are commercial software for panoramic video generation.

Multi-camera capture system

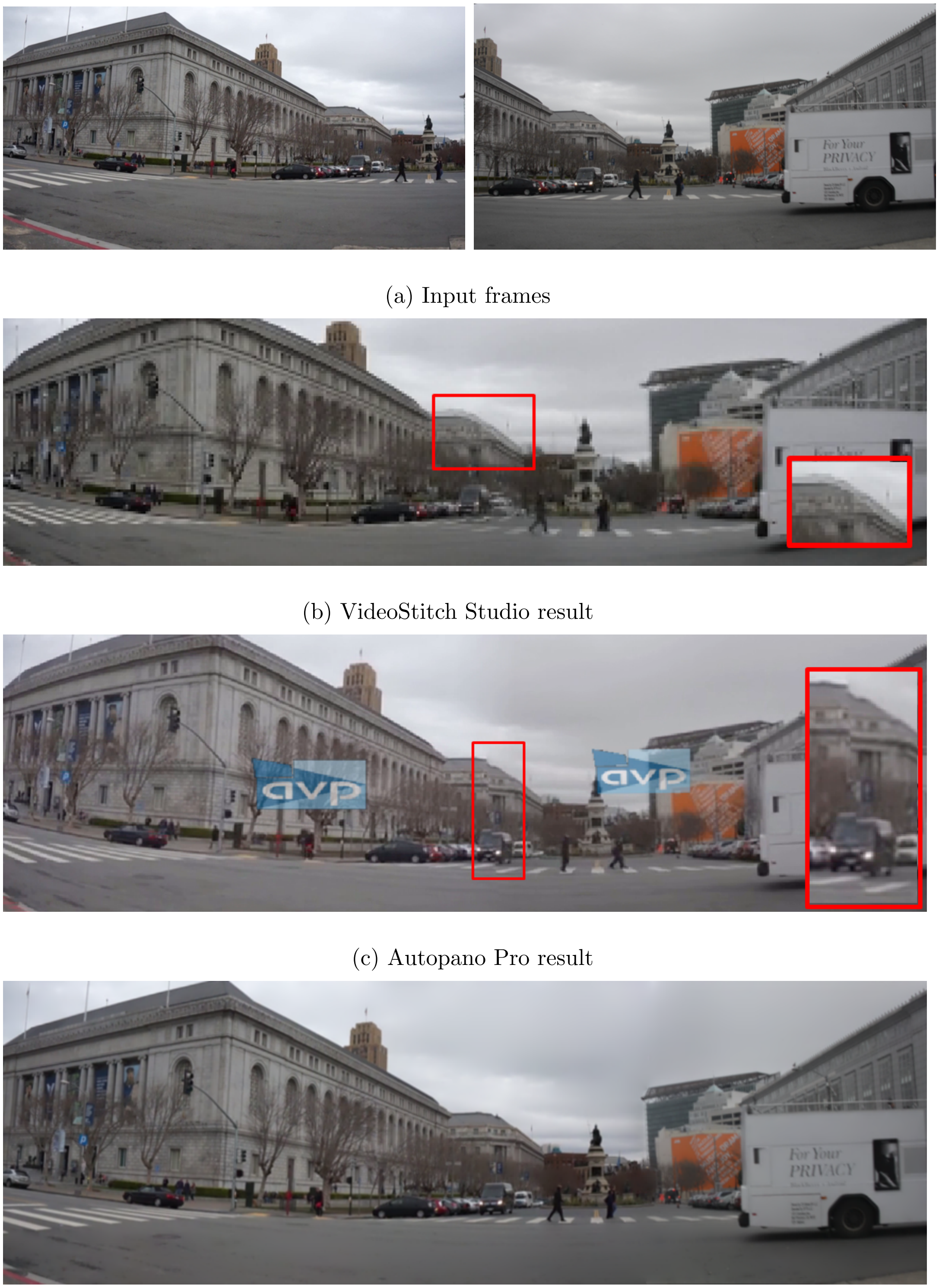
Two cameras were mounted using a dual camera bracket: one camera is Nikon D800, and the other one is Sony NEX3N. Our video stitching technique allows a flexible multi-camera system setup, and camera models can be replaced with other types (refer to Figure 5.1 for our camera capture system).

Results

Figure 5.8 shows comparisons between two video stitching methods. The input videos are taken from GCW dataset [38]. Autopano Pro cannot consistently align two frames well, therefore the frame stitching result suffers from broken structure artifacts, as shown in Figure 5.8(b). Both the building and the advertisement sign have been broken into two parts. On the other hand, our result is free from any artifact as shown in Figure 5.8(c).

Figure 5.9 shows comparisons among three video stitching methods. VideoStitch Studio cannot consistently align two frames well, and therefore the frame stitching result suffers from ghosting artifact, as shown in the red rectangle in Figure 5.9(b). Autopano Pro has the same problem, and thus it leads to broken structure artifacts. Both the building and the car have been broken into two parts in the stitching result, as indicated in Figure 5.9(c). On the other hand, our result is free from any artifact, as shown in Figure 5.9(d).

Figure 5.10 shows comparisons between two video stitching methods. The input videos are taken from GCW dataset [38]. VideoStitch Studio cannot consistently well align two frames, and ghosting artifacts have been introduced into the stitching result, as shown in Figure 5.10(b). The road lines are duplicated due to



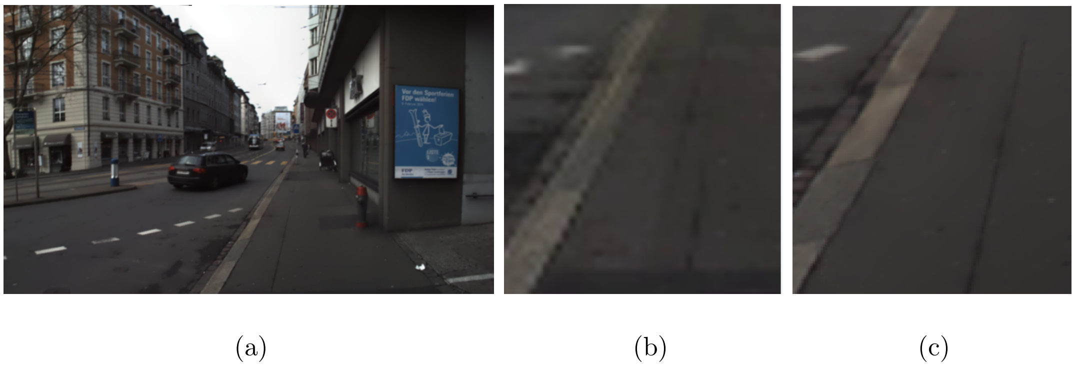
(a) Input frames

(b) VideoStitch Studio result

(c) Autopano Pro result

(d) Our result

Figure 5.9: Comparisons among three video stitching methods.



(a) (b) (c)

Figure 5.10: Comparisons between two video stitching methods. From left to right, the close-ups show comparisons to the methods: (b) VideoStitch Studio, (c) our result. VideoStitch Studio introduces misalignment into the results, result in ghosting artifacts. No misalignment occurs in our result.

misalignment. On the other hand, Our result is free from any artifact as shown in Figure 5.10(c).

5.3.7 Discussion

Our method relies on the optical flow method to compute the motion field, which are time-consuming and unreliable; therefore, a confidence map was incorporated for the optical flow method to compensate motion estimation errors so that inaccurate motion vectors would have a lower weight in affecting the warping strength. However, the side effect of such a method is that some regions with many inaccurate motion vectors will completely lose control on the warping step, and this would lead to distorted results. To reduce the effects of accumulated motion estimation and frame warping errors, semi-independent frames are used for resetting the accumulated errors to zero every few frame. Nevertheless, semi-independent frame warping can only warp the frame as similar as possible to the previous stitching result. Minor differences can be observed across semi-independent results and their previous frame stitching result. Since the video frame warping is dependent on previous stitching result, a poor stitching result would affect all the subsequent frame stitching.

5.4 FEATURE TRAJECTORY GUIDED VIDEO STITCHING

The motion map guided stitching method works well on a range of examples. However, this approach highly relies on dense motion maps to calculate frame warping functions. Optical flow based motion estimation methods often introduce errors into motion maps in textureless regions or regions with significant moving objects. Furthermore, this approach depends on previous frame stitching results to stitch successive frames, which accumulates errors across frames. Finally, this approach is slow as frame dependency makes parallel processing impossible. To stitch videos in a more reliable and faster way, instead of using motion maps as the stitching guidance, we develop a method that uses sparse feature trajectory as the guidance. To do this, we first estimate the target motion trajectory for the stitched video output. Then we warp each frame view according to the trajectory guidance. Finally, we blend the warped views together to get the stitched frames. The key components of this algorithm are discussed below.

5.4.1 Target camera motion trajectory estimation

Dense motion map or sparse motion vector?

To obtain a temporal coherent output panoramic video, it is necessary to stitch every set of input video frames in a consistent manner. This is usually achieved by using a target camera motion path as the guidance so that frames are consistently warped and stitched together. A straightforward way to create a target camera motion trajectory is to use the dense motion maps of the input videos as what was done in the previous method. However, motion maps cannot be used directly; the motion maps of each input video must be stitched in order to create a panoramic motion map as the guidance. This step introduces stitching dependency since motion map stitching is based on the seam finding result of the previous frame stitching. Getting consistent seam finding results is extremely difficult for video stitching. Although our improved graph cut seam finding method maintains certain consistency, it tends to be brittle when input videos contain fast moving objects. In addition, the dependency on previous seam finding results introduces accuracy and efficiency issues. Warping and stitching errors accumulated across frames. No parallel processing can be incorporated into video stitching.

Due to the above reasons, sparse motion vectors are used instead of dense motion maps for creating target camera motion trajectory. With SIFT feature detection and KLT tracking algorithms, we can get the motion trajectories of the sparse features for each individual input video. Then a desired camera motion trajectory for the output panoramic video can be generated using such sparse feature motion trajectories. The next question is how to generate the desired camera motion trajectory for the output video before it is even generated. To address this problem, it is necessary to first discuss the parallax removal issue in video stitching.

Parallax removal

A reasonable stitching result requires that the two neighboring views have wide enough overlapping region. For this reason, we set the baseline for two neighboring cameras to be less than 12 cm. Parallax would be introduced into the neighboring views with such separation of two cameras. Cylindrical/spherical projection or homography cannot account for such parallax, and the stitching result will suffer from ghosting artifacts or broken structures. Local homography based methods such as parallax-tolerant image stitching can handle parallax well, but this method relies on seam finding to obtain reasonable results. However, inconsistent seam finding creates issues for maintaining temporal coherence in video stitching tasks. Therefore, we decide to avoid using a stitching method based on seam finding to perform individual frame stitching. As a side benefit of having an inter-camera configuration that is “less than 12 cm”, there would be no inputs with extensive parallax as the examples in parallax-tolerant image stitching. Thus, it would be possible to eliminate the parallax between neighboring views with proper image warping.

Specifically, we first estimate feature correspondences between two neighboring views. We then need to account for parallax and align the neighboring views together. The alignment model chosen to handle parallax in a better way is global homograph with content-preserving warping. We use global homography to globally and roughly align the whole overlapping area, and then use content-preserving warping to refine the alignment result. To align the neighboring views, a direct way is to apply the warping function on only one view so that it can match with the other. This approach can use the estimated feature correspondences directly with no extra computation. However, it forces one view to warp significantly to match up with the other to eliminate the parallax. This introduces imbalanced warping effects which could cause misalignment in regions with large displacement. We address this problem by conducting warping operations on both views. We first estimate the in-between middle positions for all feature correspondences. We then warp both views to match up with the in-between middle virtual view to reach the alignment. Our solution warps both views with equal strength, and thus can eliminate parallax and align images in a better way.

Afterwards, we create the ideal target camera motion trajectory for the output panoramic video with the above idea. Starting from frame 1, we first estimate feature correspondences between two neighboring views. Following this step, we estimate the in-between middle positions for all corresponding features and use them as the ideal motion position for frame 1. To obtain motion trajectories with temporal coherence, we use KLT tracking to find the corresponding feature points in the successive frames and get middle positions. Certain image content would disappear and new image content would be added into the video frames during camera motion. To ensure consistent control for all image regions, we estimate SIFT feature correspondences for each individual frame pairs and continually adding the features that lie on the newly added region.

5.4.2 Frame stitching

After obtaining the target camera motion trajectory, we can then warp each individual frame and stitch them together. We use global homograph with content-preserving warping as the alignment model. To remove perspective distortion and reduce the warping effect as much as possible, we first project all views onto a cylindrical surface to roughly align the images. We describe the detailed frame warping steps below.

For simplicity, we consider the task of stitching two input videos V_l and V_r. More videos can be stitched in the same way. We use f_l^1 and f_r^1 to denote the first frame of two input videos.

Data term. The feature points in frame f_l^1 and f_r^1 should be moved to match their in-between middle target positions in the virtual frame so that they can be well aligned. Since a feature point P^j is not usually coincident with any mesh vertex, we find the mesh cell that contains P^j and use a linear combination of the four cell vertices to represent it. The linear combination coefficients are computed using the inverse bilinear interpolation method [20]. These coefficients are then used to combine the vertices in the warped frame $\hat{f}_l^1$ to compute $\hat{P}^j$. We can then define the alignment term as follows.

$$E_d = \sum_{j=1}^{n} \| \sum \alpha_{j,k} \hat{V}_{j,k} - \tilde{P}^j \|^2, \tilde{P}^j = (P_l^j + P_r^j)/2$$

where n is the number of feature points, $\alpha_{j,k}$ is the bilinear combination coefficient, and $\hat{V}_{j,k}$ is a vertex of the mesh cell that contains $\hat{P}_j$, and $\tilde{P}^j$ is the target feature point in the in-between middle position of the virtual frame.

Smoothness term. To further minimize the local distortion during warping, we encourage each mesh cell to undergo a similarity transformation. We use the quadratic energy term from [23] to encode the similarity transformation constraint. Specifically, consider a triangle $\triangle \bar{V}_1 \bar{V}_2 \bar{V}_3$. Its vertex $\bar{V}_1$ can be represented by the other two vertices as follows,

$$\bar{V}_1 = \bar{V}_2 + u(\bar{V}_3 - \bar{V}_2) + vR(\bar{V}_3 - \bar{V}_2), R = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \tag{5.5}$$

where u and v are the coordinates of $\bar{V}_1$ in the local coordinate system defined by $\bar{V}_2$ and $\bar{V}_3$. If this triangle undergoes a similarity transformation, its coordinates in the local coordinate system will not be changed. Therefore, the similarity transformation term can be defined as follows,

$$E_s(\hat{V}_i) = w_s \| \hat{V}_1 - (\hat{V}_2 + u(\hat{V}_3 - \hat{V}_2) + vR(\hat{V}_3 - \hat{V}_2)) \|^2, \tag{5.6}$$

where u and v are computed from Equation 5.5. We sum $E_s(\hat{V}_i)$ over all the vertices to obtain the full smoothness energy term E_s. Here w_s measures the saliency value of the triangle $\triangle \bar{V}_1 \bar{V}_2 \bar{V}_3$ using the same method as [31]. We use this saliency weight to distribute more distortion to less salient regions than those salient ones.

Global alignment term. The data term above only directly constrains warping of the overlapping image region with feature points. For other regions, content-preserving warping often distorts them. To solve this problem, we first estimate the best-fitting homography according to the control points and then employ this best-fitting homography to globally pre-warp the frame. As the pre-warping result often provides a good approximation, our method encourages the regions without control points to be as close to the pre-warping result as possible. We therefore define the following global alignment term,

$$E_g = \sum_i \tau_i \|\hat{V}_i - \bar{V}_i\|^2, \tag{5.7}$$

where $\hat{V}_i$ and $\bar{V}_i$ are the corresponding vertex in the content-preserving warping result and in the pre-warping result. τ_i is a binary value. We set it to 1 if there is no feature point in the neighborhood of V_i; otherwise it is 0. This use of τ_i provides flexibility for local alignment.

Optimization. We combine the above three energy terms into the following energy minimization problem,

$$E = \alpha E_d + \beta E_g + \gamma E_s, \tag{5.8}$$

where α, β, and γ are the weight of each term with default values 1.0, 0.7, and 0.3, respectively. The above minimization problem is quadratic and is solved using a standard sparse linear solver. Once we obtain the output mesh, we use texture mapping to render the final result.

5.4.3 Further temporal coherence improvement

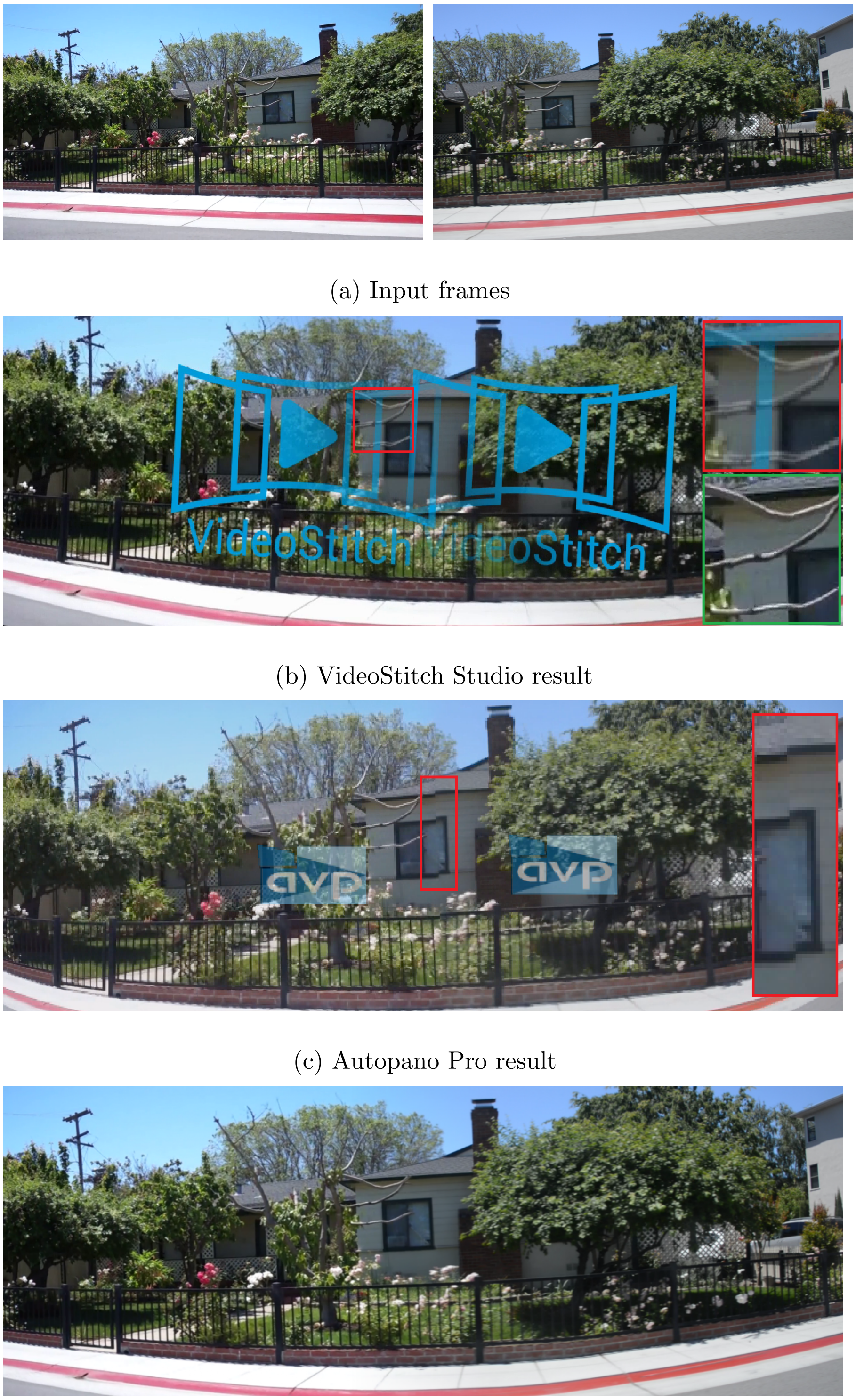
The above method usually can generate reasonable results. However, the feature set for each frame could be different from its successive frame set due to camera motion, which could cause the global alignment that is estimated from the feature set to be different across frames. This could result in temporal incoherent frame transition in the non-overlapping area. To compensate for this incoherence, we estimate the global alignment for the first frame set, and then use the same global alignment for the rest of the frames. This global alignment approximation can perform well enough because we have already used cylindrical projection to pre-align all input video frames and remove major horizontal and vertical misalignment. Under such a condition, the frame warping in the non-overlapping region can be propagated into the rest of the frames to keep the temporal coherence.

5.4.4 Experiments

We use the same camera array setup as described in the last section for the feature trajectory guided video stitching. We experimented our method on several challenging videos. Figure 5.11 shows frame results of three panoramic videos generated from three different stitching algorithms: (b) VideoStitch Studio, (c) Autopano Pro and (d) our result. VideoStitch Studio has severe ghosting artifacts around the tree area, as indicated by the red rectangle in Figure 5.11(b). Autopano Pro result suffers from broken structure artifacts around the window region, as indicated by the red rectangle in Figure 5.11(c). Our result is free of artifacts, as shown in Figure 5.11(d).

Figure 5.12 shows another stitching result comparison. VideoStitch Studio introduces ghosting artifacts around the house ceiling area, as indicated by the red rectangle in Figure 5.12(b). Autopano Pro breaks the wooden panel, as indicated by the red rectangle in Figure 5.11(c). Our result is free of any artifact, as shown in Figure 5.12(d).

Our algorithm can also be extended to handle multi-camera video stitching



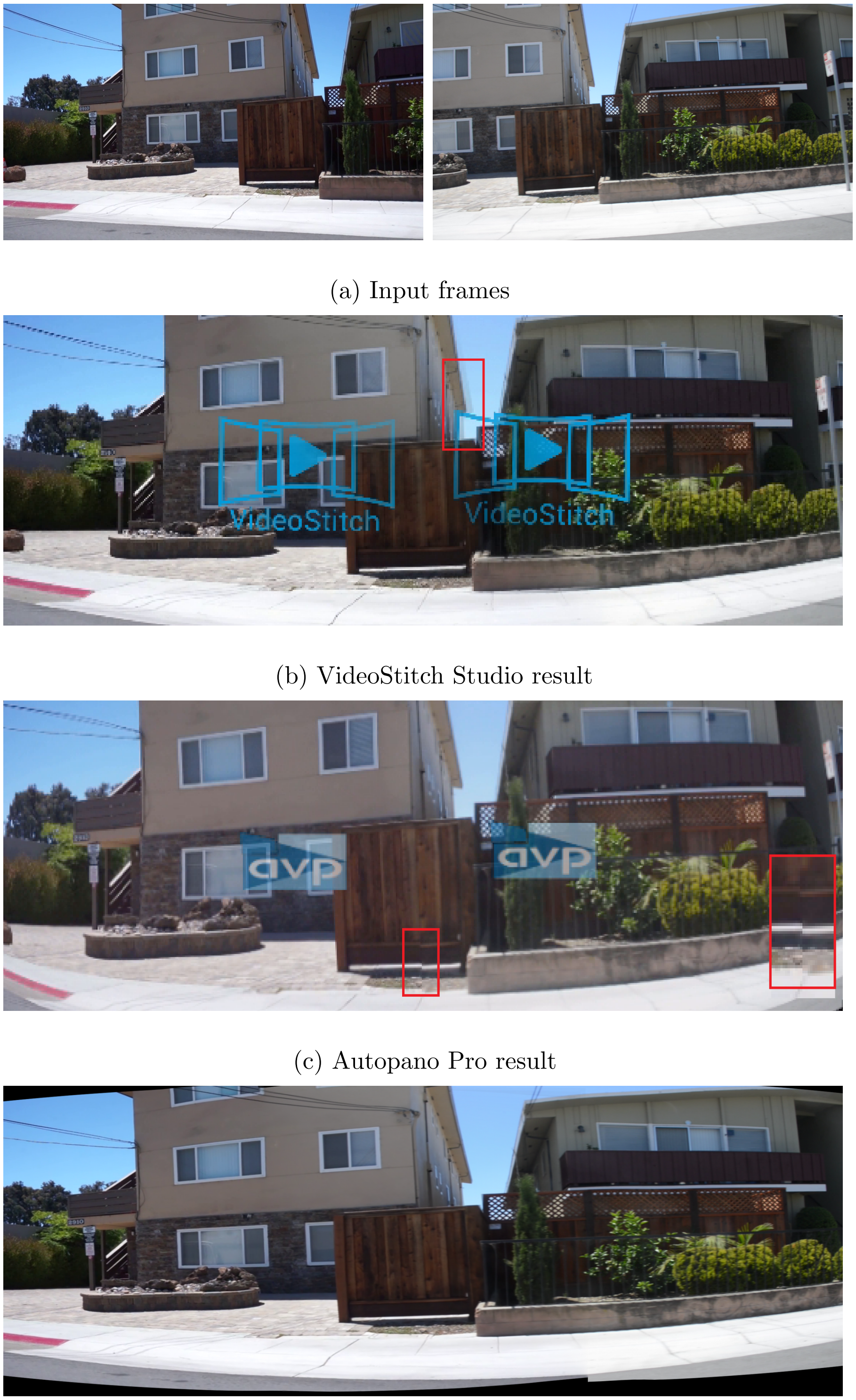
(a) Input frames

(b) VideoStitch Studio result

(c) Autopano Pro result

(d) Our result

Figure 5.11: Comparisons among three video stitching methods.



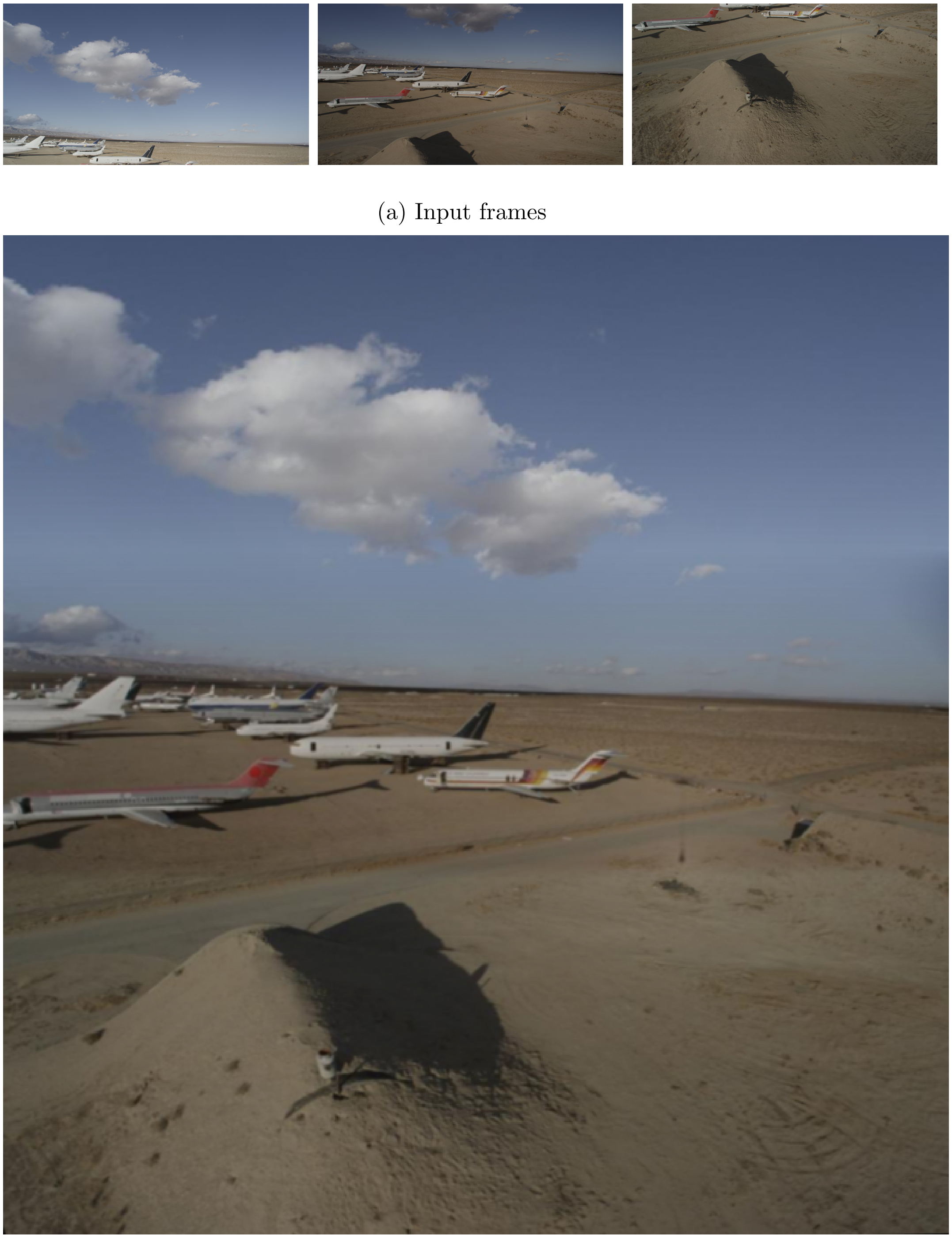
(a) Input frames

(b) VideoStitch Studio result

(c) Autopano Pro result

(d) Our result

Figure 5.12: Comparisons among three video stitching methods.



(a) Input frames

(b) Our result

Figure 5.13: Multi-video stitching result.

tasks. Figure 5.13 shows an example of stitching three input videos. The input videos are taken from GCW dataset [38].

5.4.5 Discussion

Our method uses sparse feature trajectories as the guidance to warp and stitch frames consistently. As a result, our method preserves all the camera movements such as rolling shutter artifacts in the input videos. However, our method can be easily extended to generate stabilized panoramic videos by smoothing the desired camera motion path before frame warping and stitching. One limitation of our method is that since our algorithm eliminates parallax by warping the neighboring two views to match with the in-between middle virtual view, we can only handle a reasonable amount of parallax. If there are objects that are in front of the camera within one meter, our algorithm could introduce ghosting artifacts into the results.

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 RESEARCH CONTRIBUTIONS

In this dissertation, we focus on the problem of stitching images and videos that existing techniques cannot handle well. We first contribute a parallax-tolerant image stitching technique to stitch 2D images with large parallax. Traditional stitching techniques that use homography-based transformations to stitch images cannot generate seamless stitching results. Our method is developed based on the observation that input images do not need to be perfectly aligned over the whole overlapping area. Instead, they only need to be aligned in a way that there exists a local region where they can be seamlessly blended together. We develop a randomized algorithm to search for a local homography, which, combined with content-preserving warping, allows for optimal stitching.

We also develop a technique for stitching stereoscopic panoramas from stereo images casually taken using a stereo camera. Stereoscopic image stitching needs to address three challenges: how to deal with parallax, how to stitch the left- and right- view panorama consistently, and how to take care of disparity during stitching. We address these challenges by first stitching the left images with the parallax-tolerant image stitching method to create the left view panorama, then stitching the disparity maps with a disparity optimization, finally warping and stitching the right images according to the optimized disparity map and the stitched left view panorama.

We then extend the image stitching problem into the video domain and present two techniques to stitch pre-synchronized videos captured from a fixed or hand-held camera array which contains multiple cameras with fixed inter-camera configurations. To generate stitched videos with temporal coherence, we first develop a dense motion map guided video stitching technique that warp frames according to target motion maps. We categorize video frames into four different frame types. Independent frames do frame warping independently; full-reference frames do frame warping based on previous frame warping result and motion field information; reduced-reference frames do frame warping based on limited previous frame warping output and motion field information; and semi-independent frames do frame warping only based on motion field information. In this way, we can stitch videos with temporal coherence. After that, we then develop a video stitching technique based on feature trajectory guidance. Such a method uses frame feature trajectories to generate a desired camera motion path and then uses global transformation with content-preserving warping to warp individual frames to match with the ideal camera motion path. Finally, we use alpha blending to blend all warped frames together to create final panoramic videos.

6.2 FUTURE DIRECTIONS

To better analyze the parallax problem for image stitching tasks, one future direction is to create a benchmark dataset for parallax related research. In such a benchmark dataset, we could provide accurate parallax measurements and stitching results for different scenes in order to analyze how parallax affects the image stitching process and improve the stitching techniques to handle parallax in a better way.

For panoramic video stitching, currently we do not incorporate video stabilization process into the video stitching technique. But the feature trajectory guided video stitching method we presented in chapter 5 can be extended to generate panoramic videos with temporally coherent and stabilized video content. This can be done by smoothing the desired camera motion trajectory to remove the wobbling artifacts before frame warping and stitching.

Another future direction is to extend the current video stitching technique into the stereoscopic video domain to create high-quality stereoscopic panoramic videos. In addition, we require users to fix the inter-camera configuration in this dissertation for the video stitching task. More flexible use cases could be explored so that users can create video panoramas in more casual ways.

REFERENCES

[1] Adobe. *Adobe Photoshop CS5 Extended*. http://www.adobe.com.

[2] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski. Photographing long scenes with multi-viewpoint panoramas. *ACM Trans. Graph.*, 25(3):853–861, 2006.

[3] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. *ACM Trans. Graph.*, 23(3):294–302, 2004.

[4] A. Agarwala, K. C. Zheng, C. Pal, M. Agrawala, M. Cohen, B. Curless, D. Salesin, and R. Szeliski. Panoramic video textures. *ACM Trans. Graph.*, 24(3):821–827, July 2005.

[5] T. Basha, Y. Moses, and S. Avidan. Stereo seam carving: a geometrically consistent approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2513–2525, 2013.

[6] K. Benko. *Panorama photo stitcher*. http://hugin.sourceforge.net/.

[7] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vision*, 74(1):59–73, 2007.

[8] P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[9] N. H. I. S. BV. *Graphical User Interface for Panorama Tools.* http://www.ptgui.com/.

[10] Camargus. *Maxx Zoom.* http://www.camargus.com/.

[11] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[12] V. Couture, M. S. Langer, and S. Roy. Panoramic stereo video textures. In *IEEE International Conference on Computer Vision*, pages 1251–1258, 2011.

[13] F. Dornaika and R. Chung. Mosaicking images with parallax. *Signal Processing: Image Communication*, 19(8):771–786, 2004.

[14] A. Eden, M. Uyttendaele, and R. Szeliski. Seamless image stitching of scenes with large motions and exposure differences. In *IEEE CVPR*, pages 2498–2505, 2006.

[15] FullView. *FullView.* http://www.fullview.com/.

[16] J. Gao, S. J. Kim, and M. S. Brown. Constructing image panoramas using dual-homography warping. In *IEEE CVPR*, pages 49–56, 2011.

[17] J. Gao, Y. Li, T.-J. Chin, and M. S. Brown. Seam-driven image stitching. In *Eurographics 2013*, pages 45–48, 2013.

[18] GoPano. *GoPano.* http://www.gopano.com/.

[19] P. Grey. *Ladybug by Point Grey.* http://wwx.ptgrey.com//Products/Ladybug5/.

[20] P. S. Heckbert. Fundamentals of texture mapping and image warping. Master's thesis, Citeseer, 1989.

[21] D. M. Hoffman, A. R. Girshick, K. Akeley, and M. S. Banks. Vergence–accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of vision*, 8(3):33–33, 2008.

[22] H.-C. Huang and Y.-P. Hung. Panoramic stereo imaging system with automatic disparity warping and seaming. *Graphical Models and Image Processing*, 60(3):196–208, 1998.

[23] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics*, 24(3):1134–1141, 2005.

[24] H. Ishiguro, M. Yamamoto, and S. Tsuji. Omni-directional stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):257–262, 1992.

[25] W. Jiang and J. Gu. Video stitching with spatial-temporal content-preserving warping. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 42–48, June 2015.

[26] Kolor. *Kolor Autopano Video Pro.* http://www.kolor.com/.

[27] P. Krähenbühl, M. Lang, A. Hornung, and M. Gross. A system for retargeting of streaming video. *ACM Trans. Graph.*, 28(5):126:1–126:10, 2009.

[28] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.*, 22(3):277–286, 2003.

[29] M. Lang, A. Hornung, O. Wang, S. Poulakos, A. Smolic, and M. Gross. Nonlinear disparity mapping for stereoscopic 3d. *ACM Transactions on Graphics*, 29(4):75:1–75:10, 2010.

[30] W.-Y. Lin, S. Liu, Y. Matsushita, T.-T. Ng, and L.-F. Cheong. Smoothly varying affine stitching. In *IEEE CVPR*, pages 345–352, 2011.

[31] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving warps for 3d video stabilization. *ACM Transactions on Graphics*, 28(3):44:1–44:9, 2009.

[32] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[33] B. Mendiburu. *3D movie making: stereoscopic digital cinema from script to screen.* CRC Press, 2009.

[34] S. Nayar. Catadioptric omnidirectional camera. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 482–488, Jun 1997.

[35] T. Pajdla, T. Svoboda, and V. Hlaváč. Panoramic vision. chapter Epipolar Geometry of Central Panoramic Catadioptric Cameras, pages 73–102. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[36] S. Peleg, M. Ben-Ezra, and Y. Pritch. Omnistereo: Panoramic stereo imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):279–290, 2001.

[37] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet. Mosaicing on adaptive manifolds. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1144–1154, 2000.

[38] F. Perazzi, A. Sorkine-Hornung, H. Zimmer, P. Kaufmann, O. Wang, S. Watson, and M. H. Gross. Panoramic video from unstructured camera arrays. *Comput. Graph. Forum*, 34(2):57–68, 2015.

[39] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, 2003.

[40] S. Pirk, M. F. Cohen, O. Deussen, M. Uyttendaele, and J. Kopf. Video enhanced gigapixel panoramas. In *SIGGRAPH Asia 2012 Technical Briefs*, SA '12, pages 7:1–7:4, New York, NY, USA, 2012. ACM.

[41] A. Rav-Acha, G. Engel, and S. Peleg. Minimal aspect distortion (mad) mosaicing of long scenes. *Int. J. Comput. Vision*, 78(2-3):187–206, 2008.

[42] A. Rav-Acha, Y. Pritch, D. Lischinski, and S. Peleg. Dynamosaics: video mosaics with non-chronological time. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 58–65 vol. 1, June 2005.

[43] C. Richardt, Y. Pritch, H. Zimmer, and A. Sorkine-Hornung. Megastereo: Constructing high-resolution stereo panoramas. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1256–1263, 2013.

[44] A. Roman and H. P. Lensch. Automatic multiperspective images. In *Eurographics Symposium on Rendering*, pages 83–92, 2006.

[45] R. Ronfard and G. Taubin. *Image and geometry processing for 3-D cinematography*, volume 5. Springer Science & Business Media, 2010.

[46] O. Schreer, I. Feldmann, C. Weissig, P. Kauff, and R. Schafer. Ultrahigh-resolution panoramic imaging for format-agnostic video production. *Proceedings of the IEEE*, 101(1):99–114, Jan 2013.

[47] S. M. Seitz, A. Kalai, and H.-Y. Shum. Omnivergent stereo. *International Journal of Computer Vision*, 48(3):159–172, 2002.

[48] S. M. Seitz and J. Kim. Multiperspective imaging. *IEEE Computer Graphics and Applications*, 23(6):16–19, 2003.

[49] H.-Y. Shum and R. Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *IEEE ICCV*, pages 953–956, 1998.

[50] H.-Y. Shum and R. Szeliski. Stereo reconstruction from multiperspective panoramas. In *IEEE International Conference on Computer Vision*, pages 14–21, 1999.

[51] V. Studio. *VideoStitch Studio.* http://www.video-stitch.com/.

[52] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2432–2439, 2010.

[53] R. Szeliski. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.*, 2(1):1–104, 2006.

[54] R. Szeliski. Image alignment and stitching: a tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2(1):1–104, 2006.

[55] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. In *ACM SIGGRAPH*, pages 251–258, 1997.

[56] J. Tompkin, F. Pece, R. Shah, S. Izadi, J. Kautz, and C. Theobalt. Video collections in panoramic contexts. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 131–140, New York, NY, USA, 2013. ACM.

[57] Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee. Optimized scale-and-stretch for image resizing. *ACM Trans. Graph.*, 27(5):118:1–118:8, 2008.

[58] L. Wolf, M. Guttmann, and D. Cohen-Or. Non-homogeneous content-driven video-retargeting. In *IEEE ICCV*, 2007.

[59] J. Yu and L. McMillan. A framework for multiperspective rendering. In *EGSR*, pages 61–68, 2004.

[60] J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter. As-projective-as-possible image stitching with moving DLT. In *IEEE CVPR*, 2013.

[61] F. Zhang and F. Liu. Parallax-tolerant image stitching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3269, 2014.

[62] Q. Zhi and J. Cooperstock. Toward dynamic image mosaic generation with robustness to parallax. *Image Processing, IEEE Transactions on*, 21(1):366–378, Jan 2012.

[63] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: the crossed-slits projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(6):741–754, 2003.