Towards Dynamic Mosaic Generation with Robustness to Parallax Effects

Zhi Qi

Doctor of Philosophy

Department of Electrical and Computer Engineering

McGill University

Montreal, Quebec

2008-09-30

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

DEDICATION

This document is dedicated to my dear grandfather for everything he gave to me.

ACKNOWLEDGEMENTS

I would like to first thank my supervisor, Jeremy R. Cooperstock, who provided me not only the financial aid necessary for the years of study, but also a lot of encouragement and instruction to help me walk steadily through this long journey. Many thanks also go to my advisory committee: Frank Ferrie, Sébastien Roy and Irene Leszkowicz, from whom I received many valuable suggestions and brilliant inspirations during each committee meeting. Appreciation also goes to Kaleem Siddiqi for the lectures, which led me into the field of computer vision.

To my friends and colleagues at CIM, especially Samuel Audet, Xu Huan, Stéphane Pelletier, Yin Jianfeng and Wang Guangyu thanks for all the help. You are good friends and wise colleagues. Extra thanks to Stéphane Pelletier for the translation of the abstract in French.

I also appreciate my family, Hong, Meimei, parents-in-law and Naiying. Thank you all for the long-lasting strong and warm support.

ABSTRACT

This thesis addresses the challenges of parallax and object motion in image mosaicing. Traditional techniques construct their panoramas through an image registration process that minimizes the differences in intensity or structure in the overlapping areas between inputs. In order to reduce the artifacts due to misregistrations caused by parallax and object motion, traditional approaches require planar content or impose the constraint that inputs be provided either by a purely rotational camera or from dense sampling of the scene. However, these solutions are often impractical or fail to address the needs of all applications.

In this thesis, we present three novel mosaicing approaches that use depth information to compensate for the above limitations. The first approach starts by synthesizing a dense set of virtual images at positions between the source cameras. Next, an appearance-based optimization procedure is applied to select a group of strips from the collection of virtual images and the real inputs to construct the final mosaic. The virtual dense sampling greatly reduces parallax effects between the real input frames, and thus results in significantly improved mosaic outputs compared to traditional image mosaicing algorithms.

The second approach formulates the image mosaicing as a view synthesis problem. It renders the panorama as a single frame using depth estimates from the view point of a virtual mosaicing camera. It notably includes the contents in non-overlapping regions between sources by using a depth propagation procedure. Provided that the depth values are reasonable, the output mosaic result is perceptually satisfactory and free of parallax-related artifacts.

The third approach extends this work to the synthesis of dynamic video mosaics, i.e., that represent dynamic events in the environment. In order to cope with the challenges of parallax and motion, our novel depth-based dynamic mosaicing technique projects the foreground and background layers separately onto the mosaicing image plane. Importantly, the resulting video mosaic preserves spatiotemporal motion consistency.

ABRÉGÉ

Cette thèse traite du problème de surmonter les difficultés causées par l'effet de parallaxe et le mouvement d'objets lors du mosaicage d'images. Les méthodes traditionnelles de mosaicage d'images produisent des panoramas en recalant des images de manière à minimiser les différences d'intensité ou de structure entre les régions superposées de ces images. Afin de réduire les défauts occasionnés par un mauvais recalage résultant de l'effet de parallaxe ou des mouvements d'objets, les méthodes traditionnelles requièrent soit une scène planaire, soit des images résultant d'une simple rotation d'une camera, soit encore un échantillonnage dense de la scène. Toutefois, ces solutions sont souvent impraticables ou ne satisfont pas les besoins de certaines applications.

Dans cette thèse, nous présentons trois nouvelles approches de mosaicage d'imagesF utilisant l'information de profondeur afin de surmonter les limites énoncées ci-haut. La première approche utilise l'échantillonnage dense virtuel afin de réduire l'effet de parallaxe entre les images acquises, ce qui résulte en une nette amélioration de la qualité des panoramas par rapport à ceux obtenus avec les techniques traditionnelles.

La deuxième approche construit le panorama correspondant à une position virtuelle de la caméra en utilisant des estimations de profondeur. Lorsque ces valeurs sont raisonnables, le résultat devrait être satisfaisant visuellement et libre de tout défaut relié à l'effet de parallaxe.

La troisième approche fait passer le mosaicage d'images de la synthèse d'une seule image à la génération d'une séquence vidéo de mosaiques capturant des évènements dynamiques se produisant dans l'environnement. Cette nouvelle technique, basée sur la profondeur, projette séparément les segmentations d'avant-plan et d'arrière-plan sur le plan de la mosaique afin de produire une séquence vidéo de mosaiques d'images, une image à la fois. Cette approche préserve la cohérence spatiale et temporelle des mouvements d'objets, ainsi que le contenu statique dans la scène.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
Introduction

This thesis introduces our efforts to create high quality mosaics given a sparse sampling of either a static or dynamic environment with potentially significant depth variance and where the camera configuration may introduce parallax. We begin with a discussion of the limitations of traditional image mosaicing methods, which motivates the research explored in this thesis.

1.1 Limitations of Traditional Image Mosaicing Methods

Image mosaicing refers to a process of combining multiple images with overlapping field-of-view (FOV) to produce a panorama. In theory, it compensates for the limited FOV of a single camera without sacrificing resolution or introducing undesirable lens distortions, as does the use of a wide angle lens, or even an omnidirectional lens, to achieve a comparable panoramic view.

Image mosaicing is widely used in applications ranging from scientific observation to consumer digital cameras. However, in order to ensure a perceptually correct result, traditional image mosaicing systems typically make one of the following assumptions:

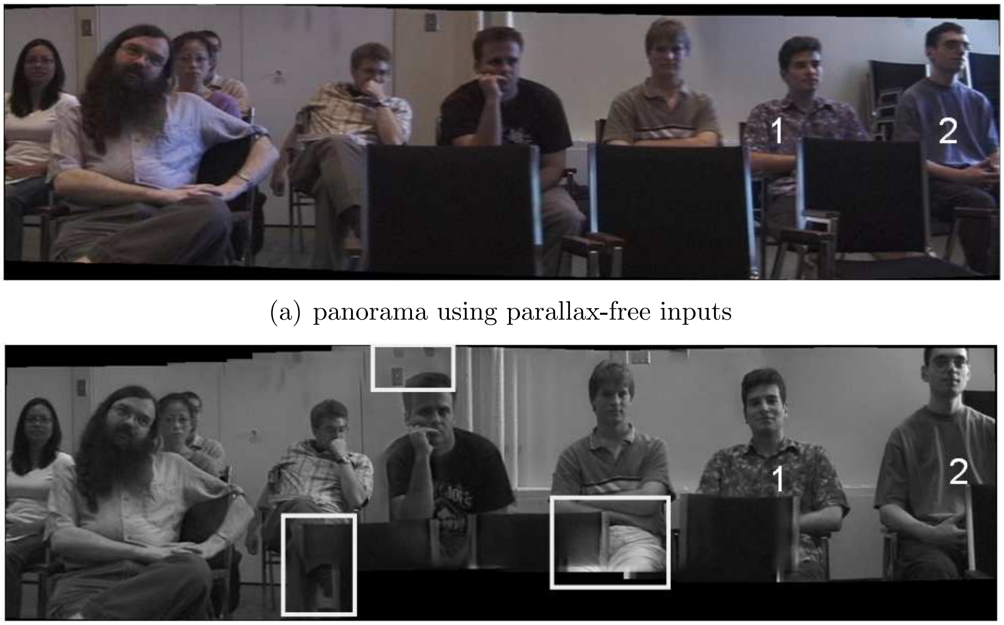
- the scene is limited in depth variance, i.e., is frontal-parallel or nearly planar;

- cameras have a parallax-free[1] configuration, such as a virtual purely rotational camera synthesized in QuicktimeVR [15];
- the acquisition system provides a dense sampling of the environment regardless of the camera motion model or the scene topology, e.g., manifold mosaicing [39][40].

These assumptions may fail to address the needs of all applications. For example, an image mosaicing system applied in an indoor environment generally has to deal with structured contents with more complex depth distributions than planar scenery. Inputs obtained by cameras with a parallax-free configuration may result in a perceptually unacceptable view of scene contents if these are close to the edges of the wide FOV of the output panorama, such as the views of 1 and 2 subjects presented in Figure 1–1a. Thus, a non-parallax-free camera configuration, which provides a frontal observation using cameras directly facing the objects of interest, sometimes may be preferred. Moreover, when multiple synchronized video cameras are used to continuously capture the events occurring in the scene, a dense sampling, such as the light field [34], becomes prohibitively expensive.

If the above assumptions required by traditional image mosaicing algorithms are violated, the parallax effects between inputs make it difficult to achieve an accurate *image registration* that is applicable for the entire scene. Resulting misalignments may cause local intensity or structural inconsistencies, which induce visual artifacts

[1] Parallax is the perceived shift of an object against a background, caused either by a change in camera position or movement of the object itself.



(a) panorama using parallax-free inputs

(b) panorama using non-parallax-free inputs

Figure 1–1: Comparison of mosaic results if given inputs with or without parallax effects. (a) The panorama built from inputs with a parallax-free camera configuration presents an observation of subjects 1 and 2 from an extreme side view (b) The panorama built from inputs of a translational camera contains artifacts within the highlighted bounding boxes.

in output mosaics, as illustrated in the highlighted bounding boxes of Figure 1–1b. In this example mosaic, built by Autostitch [13] with non-parallax-free inputs, the multiple instances of the light switch on the background wall are considered *duplication errors*, and the gradual merging of human legs into the chairs of the foreground are *ghost artifacts*. Consequently, when applied to an environment where parallax is unavoidable, traditional image mosaicing techniques may fail to produce acceptable results.

The presence of moving objects in the scene is another issue that frustrates traditional image mosaicing. Dynamic contents result in inter-frame changes of texture or of features used to align images for generating the mosaic. These object motions induce *jitter* in the appearance of the background and discontinuities of object movements on a frame-by-frame basis in the results of traditional image mosaicing techniques.

This thesis addresses construction of mosaics given inputs containing both parallax effects and moving objects, which are serious challenges to current state of the art image mosaicing techniques. We observe that a reasonable image mosaicing result should satisfy the following properties:

- **Feature or structure preservation**. The final mosaic result should not violate existing features or create new features[2] in the scene; otherwise artifacts, such as ghost or duplication errors, may result, as illustrated in Figure 1–1b.
- **Shape preservation**. Objects in the mosaic result should be free of distortion, i.e., maintain the same shapes as in the inputs. Samples of distortion errors in mosaicing results can be observed in Figure 3–4.
- **Motion preservation**. If the scene contains dynamic objects, the motion of both foreground and background contents in the mosaic video should be

[2] Feature, i.e., structure, is a general concept in computer vision. It can be a group of points, edges or complex structures representing objects. The choice of features in a particular computer vision system is highly problem dependent.

consistent with that which occurs in the input video. Samples of motion inconsistency errors of static background in mosaicing results can be observed in Figure 5–7.

1.2 Thesis Overview

Following this introduction and the literature review in Chapter 2, we present three methods that exploit geometric information in the form of depth cues in order to overcome the issues of parallax and dynamic objects in the scene.

Given a pair of stereo images and their associated camera parameters, our first approach, described in Chapter 3, starts by synthesizing a dense set of virtual images at positions between the source cameras. Next, an appearance-based optimization procedure is applied to select a group of strips from the collection of virtual samples and the real inputs to construct the final mosaic. The virtual dense sampling greatly reduces parallax effects between the real input frames, and thus results in significantly improved mosaic outputs compared to traditional image mosaicing algorithms.

The second approach, discussed in Chapter 4, renders an improved panorama as a single frame, using depth estimates from the position of a virtual mosaicing camera. Traditional image mosaicing techniques perform $2D$ image registration that minimizes the intensity or structure differences in the *overlapping areas* between inputs. However, our new depth-based image mosaicing approach also preserves an appearance of smooth connections in the *non-overlapping regions.* Provided that the depth values are reasonable, the output mosaic result is perceptually satisfactory and free of parallax-related artifacts.

Our third approach, described in Chapter 5, falls into the category of *dynamic mosaics* [25], extending the problem from that of generating a single panoramic image to a *mosaic video* that captures dynamic events taking place in the environment.

In order to cope with the problems of both parallax and motion in inputs, our novel depth-based image mosaicing technique projects the segmented foreground and background layers separately onto the mosaicing image plane to construct the output mosaic video on a frame-by-frame basis. The result preserves both the temporal and spatial coherence of object motion, as well as static contents, in the scene.

Finally, Chapter 6 concludes with a discussion of future directions of research and a summary of the work in this thesis.

1.3 Contributions

This thesis makes the following contributions:

1. A novel formulation of generating image mosaicing, instead of traditionally using the image warping and stitching, as a view synthesis problem that takes advantage of $3D$ information inferred from the input cameras.
2. A validated depth-based image mosaicing algorithm that renders the panorama as the output of a virtual camera, by explicitly estimating depth values for the entire scene, when given the calibration results of input cameras.
3. The creation of a smooth motion perception criterion, which helps to generate dynamic mosaicing results that preserve not only the perception of correct motion but also the perception of motion consistency in the spatiotemporal domain.

4. The first validated dynamic mosaicing algorithm, which exhibits robustness to both parallax and object motion by virtue of its layered depth-based rendering approach and the usage of the smooth motion perception criterion.
5. A depth propagation procedure that spreads depth information from regions of reliable estimates, into neighboring areas where such information is not available, according to different criterion for either static or dynamic contents in the scene.
6. A successful implementation, which integrates image-based rendering and a smooth appearance connection-based optimization algorithm, to produce high-quality panoramas with multiple perspective projections, when given only sparse input samples.

CHAPTER 2
Background knowledge and Literature Review

In this chapter, we first briefly introduce the general background of image mosaicing, then summarize the literature related to algorithms in this field.

2.1 Background Knowledge

The image mosaicing procedure generally includes three steps. First, we register input images by estimating the *homography*, which relates pixels in one frame to their corresponding pixels in another frame. Second, we warp input frames according to the estimated homographies so that their *overlapping regions* align. Finally, we paste the warped images and blend them on a common mosaicing surface to build the panorama result.

2.1.1 Motion Models

Before introducing the procedure of image alignment, we must discuss the notation of motion models, which describe the mapping from pixels in one image to their correspondences in other images. When cameras observe a planar scene or if they undergo pure rotation, the relationship between corresponding $2D$ projections of the same $3D$ point across different images can be described precisely by motion models, also known as 3×3 *homography* matrices.

As illustrated in Table 2–1, among the five basic $2D$ motion models, the six-parameter affine matrix and the eight-parameter projective matrix are most commonly used for image mosaicing. Suppose we have corresponding homogeneous pixel vectors $\mathbf{x} = (x, y, 1)$ and $\mathbf{x}' = (x', y', 1)$ in different input images. The mapping between them is described as:

$$\mathbf{x}' \sim H\mathbf{x} \tag{2.1}$$

where H is the 3×3 homography matrix.

Group	**Projection Matrix**	**Degrees of Freedom**	**Principal Invariant**
Projective	$[\ H\ ]_{3\times3}$	8	straight lines
Affine	$[\ A\ ]_{2\times3}$	6	parallelism
Similarity	$[\ sR \mid t\ ]_{2\times3}$	4	angles
Rigid (Euclidean)	$[\ R \mid t\ ]_{2\times3}$	3	length
Translation	$[\ I \mid t\ ]_{2\times3}$	2	orientation

Table 2–1: Hierarchy of 2D coordinate transformations.

2.1.2 Direct and Feature-based Image Alignment

Image alignment, the first step of image mosaicing, estimates homography parameters by minimizing an *error metric* that measures the agreement between correspondences across input images. The error metric can either be, the *sum of squared differences* (SSD), denoted e, its robust function, $\rho(e)$ [55], which deals with outliers of correspondences, or a more complex version, as in the work of Baker and Matthews [5], which additionally models the bias and gain variances between images being compared.

If using the robust function $\rho(e)$ as the error metric, estimation of the homography parameters is defined as follows:

$$E(p + \triangle p) = \sum_{i \in \varphi} \rho([I_1(W(x_i; p + \triangle p)) - I_0(x_i)]^2) \tag{2.2}$$

Let $W(x_i; p + \triangle p)$ denote the transformation by Hx in Equation 2.1, where $\mathbf{p} = (p_1, ...p_n)$ is a vector of homography parameters. The mapping $W(x_i; p + \triangle p)$ projects pixel x_i in image I_0 into its matching pixel in image I_1. The algorithm iteratively updates p by $\triangle p$ in the direction minimizing the error metric until p converges.

Image alignment approaches include *direct methods*, which depend on the agreement between pixels in overlapping regions, as opposed to *feature-based methods*, which consider only the correspondences between salient feature points [36][37]. Accordingly, φ in Equation 2.2 represents either a set of matching pixels or a smaller set of corresponding feature points. The direct methods take advantage of the complete image information in overlapping regions and thus may generate a more reliable homography estimation than feature-based methods. This is particularly the case when dealing with textureless regions where the number of feature points is limited, or over-textured regions where the distribution of features is much denser in contrast with the remaining parts of the image.

2.1.3 Pixel Blending

Once the correspondences between input images have been correctly aligned, inputs are warped onto the common mosaicing image surface according to the estimated homographies and then merged to build the output panorama. However, due to exposure differences, misregistrations or even movement of objects in the scene, merging warped inputs is not simply an averaging process between overlapping pixels.

A better approach is to take a weighted averaging that assigns pixels closer to the center of the image higher weights before blending them. Such a technique of blending pixels by a weighted averaging is called feathering [56][60], and is helpful in overcoming the exposure differences between inputs. When integrated with the *high dynamic range*(HDR) radiance map [17] and the exposure invariant feature-based image alignment method, it can even construct panoramas over tremendous exposure differences [20]. Feathering can be performed within pixel color spaces, or in the *gradient domain* [2][33].

Recently, Jia and Tang [29] presented a seamless image stitching system using structure deformation. Based on the matching of features, the algorithm first partitioned inputs subject to the constraints of intensity coherence and structure continuity. It then deformed and propagated the features across the partitions to achieve smooth stitching of inputs. Their work built improved mosaic results over conventional feathering algorithms, especially when inputs contained intensity inconsistencies or structure misalignments.

Although the advanced blending techniques above successfully compensate for exposure variance, intensity inconsistency or a limited amount of registration error in sources, the ghost errors resulting from parallax or object motion remain beyond the capabilities of these techniques.

2.2 Literature Review: Image Mosaicing Techniques

Considerable effort has been invested to increase the robustness of mosaic results to variation of illumination, exposure variance, lens distortion, and other such challenges. For the purpose of this thesis, we only discuss image mosaicing techniques that focus on the issues of parallax and object motion in the scene.

These algorithms are categorized into three groups. The first group, summarized in Section 2.2.1, applies a parallax-free constraint that only accepts inputs of either planar scenes or arbitrary scenes observed by cameras with a parallax-free configuration. The second group, which makes use of dense sampling to compensate for depth variance in the environment, and thus offers more robustness to input camera motion, is reviewed in Section 2.2.2. The last group, involving dynamic mosaicing techniques, is discussed in Section 2.2.3.

2.2.1 Mosaicing with Parallax-free Inputs

Chen introduced *QuickTime VR* [15], which generated a panoramic view of the environment based on images taken by a purely rotating camera. The algorithm used a correlation-based image registration method and a simple average blending to build panoramas. Neighboring input images were required to have 50% overlap with each other. The factors that contributed to stitching failures of the algorithm included extreme changes of intensity, slight movement of camera center during acquisition, and object motion in the scene.

Unlike QuickTime VR, **Szeliski and Shum** [56] presented an approach to create full-view panoramic mosaics from image sequences taken by hand-held cameras, and which support more flexible camera motions other than pure rotations. However, scene contents were assumed to be far from the camera center, and appear as nearly planar. Their algorithm not only developed the estimation and refinement of camera focal lengths, but also presented a patch-based image registration to quickly align inputs for calculation of the transformation matrices between input images and one reference camera.

In order to cope with the problem of accumulated image registration errors when constructing an image mosaic from a long sequence of inputs, **Shum and**

Szeliski [52] introduced the algorithm of *global and local image alignments.* The global alignment method simultaneously updated all frame poses (rotations and focal lengths) minimizing the sum of registration errors between all matching features over the entire sequence of input images. A local alignment technique warped small image patches onto the mosaicing plane according to the result of pairwise local image registration, so that ghost errors due to small amounts of parallax introduced by object movement or lens distortion were largely removed. The bundled global and local alignment techniques significantly improved the quality of image mosaicing output.

Although successful image stitching algorithms, which made use of parallax-free inputs, had become available, the challenges of mosaic artifacts due to object motion and exposure differences were not addressed until the work by **Uyttebdaele et al.** [60]. Their approach first identified regions that contain moving objects in input images as nodes in a graph, then applied a *vertex covering algorithm* to selectively remove all but one instance of each object. Finally, it constructed a static mosaic result of the entire scene with each moving object appearing at one specific spatiotemporal position, without introducing ghost errors. However it does not respect the chronological order of object motions. The second contribution of this paper was its compensation for the exposure variance between inputs by a block-based intensity adjustment. Together these contributed to automatic image mosaicing results that exhibit far fewer artifacts than any image mosaicing algorithms contemporary to their work.

Brown and Lowe [13] described a fully automatic construction of panoramas, which was insensitive to the ordering, orientation, scale and illumination of input images. The algorithm first applied *scale invariant feature transform (SIFT)* features [36] and a probabilistic model to verify image matches so that it recognized multiple panoramas in unordered image sets. It then used a global alignment [52] to register inputs, and finally, implemented a *multi-band seamless blending* [14] to produce a panorama with smooth appearance. This contributed to a widely applicable automatic image mosaicing software, Autostitch [12].

2.2.2 Manifold Mosaicing with Dense Sampling of Inputs

Image mosaicing algorithms, which make use of dense samples of the scene, are also known as manifold mosaicing algorithms. Following the definition by **Peleg et al.** [39][40], a manifold mosaic, i.e., a multiperspective image [44], was built by projecting warped thin strips from input images onto the mosaicing surface.

A manifold mosaic can be generated by a pushbroom camera projection [22], which used a $1D$ sensor array to collect thin strips while sweeping the scene along a continuous path, as illustrated in Figure 2–1. In order to generate reasonable mosaic results, the warped strips from the inputs should present approximately uniform *optical flow* in a parallel direction and of equal magnitude. Thus, the shape of the strip depends on the type of camera motion and the width of the strip is proportional to its speed.

The use of dense sampling by manifold mosaicing algorithms overcomes the constraint of parallax-free camera configurations, and is thus capable of handling more general cases of input camera motion.

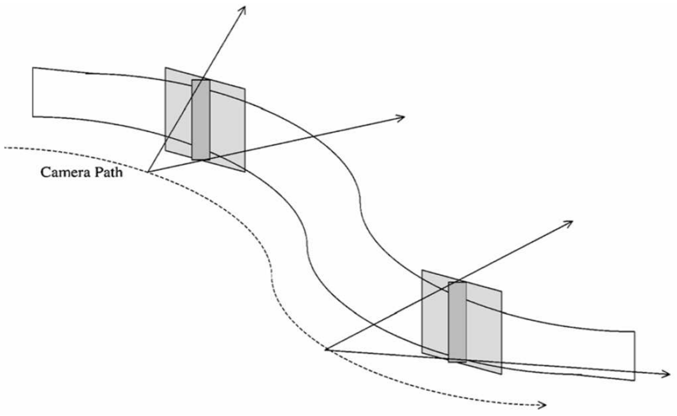


Figure 2–1: A manifold mosaic with a pushbroom camera model is built by connecting thin strips from inputs taken when the camera moves along a continuous path as the dotted line in the figure. Figure 4 of Peleg et al. [40] (©[2000]IEEE, reproduced here with permission).

Zomet et al. [68] introduced a new manifold mosaicing algorithm based on *crossed-slits projection*. All the rays in this projection model pass through two non-parallel slits: the camera path and the second line perpendicular to the camera path, as illustrated in Figure 2–2. This projection is a superset of the perspective and pushbroom projections and thus, the mosaic result can be converted easily to either of the other two projection models. Crossed-slits projection mosaicing offers the benefit that the results are closer to perspective images than those of traditional pushbroom mosaics.

Theoretically, manifold mosaicing, due to its dense sampling, exhibits greater robustness to parallax effects than other techniques summarized in the previous section. Nevertheless, it still produces noticeable artifacts in the mosaic result when presented with a complicated depth distribution or object motion in its inputs.

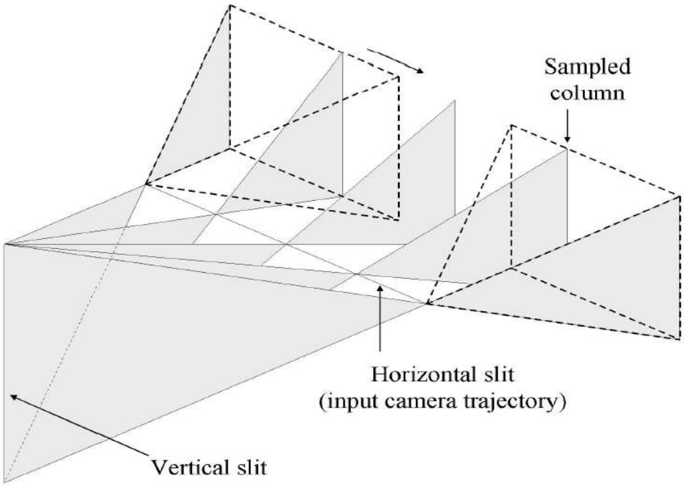


Figure 2–2: A cross slits camera projection is performed by requiring all the rays pass through two non-parallel slits: the camera path and the second line perpendicular to the camera path. Figure 3 in Zomet et al. [68] (©[2003]IEEE, reproduced here with permission).

A recent effort of **Agarwala et al.** [1] reduced the artifacts of manifold mosaicing by minimizing error functions based on criterion of smooth and continuous strip connections. This approach was applied to the production of long (wide) manifold panoramas of approximately planar scenes based on relatively sparse samples obtained by a moving hand-held camera. Instead of combining strips with regular shapes as done by Peleg et al. [39][40], they used *Markov Random Field optimization* to construct the panorama from arbitrary shaped regions of sources, so that the output exhibited the best stitching results with minimal artifacts. However, in the case of inputs containing obvious parallax, their work cannot generate a perceptually valid panorama.

Zheng et al. [67] also used dense sampling and multiperspective projections. Although their technique did not restrict itself to the use of thin strips from input images, it nevertheless shares many of the attributes of other manifold mosaicing algorithms, and so, is included here.

Each pixel in the overlapping regions between sources was registered into a layered multiperspective cylindrical space based on its estimated depth value to construct a *layered depth panorama (LDP)*. Similar to the *layered depth image* (LDI) [50], the result may include multiple pixels at different depths along each line of sight, wherein the depth is determined by the intersection of the ray and objects along its path. Integration of depth information in this manner produced reasonable panoramas, provided that the sampling rate was sufficient to guarantee that the entire scene was covered within the overlapping regions between inputs.

2.2.3 Dynamic Mosaicing

The algorithms described in the previous two sections are intended for the construction of panoramas of mainly static scenes. In this section, we introduce dynamic mosaic techniques, which address the added challenge of object motion and generate mosaic video that captures dynamic events taking place in the environment.

These algorithms apply robust image alignment procedures to minimize the impact of dynamic objects so that they can accurately estimate camera motion based only on the static contents of the scene. The static contents together with, if applicable, the dynamic contents of the same input frame are warped according to the estimated camera motion model to build the dynamic mosaicing video on a frame-by-frame basis. The much improved image registration results constitute the crucial advance of these mosaicing techniques, allowing them to cope with object motion, rather than being limited to static scenes. However, these algorithms only worked with inputs of rotational (panning) camera movements and thus, largely motivated the problem of parallax effects. Non-parallax-free camera configurations remain beyond the capabilities of these algorithms.

Irani et al. [25] defined the concepts of *static* and *dynamic* mosaics, the former as a single view of the full scene over the entire input video sequence. Any objects exhibiting motion against the static background are either excluded from the result or appear as ghost errors in the panorama. A dynamic mosaic records chronological object activities as a sequence of mosaic images, each updated according to the most recent frame in the input sequence. Again, however, the process of synthesizing these mosaics was restricted to inputs from cameras with pure panning rotations.

An important advance for the synthesis of dynamic mosaics was the work of **Sawhney and Ayer** [46], who introduced a motion-separated layered representation of the input video. The input sequence was divided based on dominant motion separation into a layer of fixed background and other layers of moving objects. Input frames were then registered based only on the static background layer, and finally these layered contents were combined into a mosaic video. Their work only analyzed inputs with panning rotations, and did not discuss other camera motions.

Bartoli et al. [6] applied a combined feature-based and direct image registration procedure to address object motion in inputs. First, a bundle adjustment based on matching features was used to generate initial estimates of camera motion models. A direct image registration was then implemented to refine these initial results locally between consecutive frames. Outliers to the motion model were classified as dynamic layer contents and the inliers as part of the static background. Although this resulted in three useful components, namely, a background panorama, a registered input sequence, and dynamic layers containing moving objects, the authors did not describe how such components could be combined to produce a true dynamic mosaic.

Taking advantage of dense sampling inputs and the manifold mosaicing synthesis technique, **Rav-Acha et al.** [4] removed the constraint of purely rotational camera movement for the construction of a dynamic video mosaic. Their algorithm first produced an aligned space-time volume based on the input video sequence. It then swept a continuous $2D$ freeform surface, or *time front*, through the space-time volume as shown in Figure 2–3 to synthesize frames constituting the final panoramic video. The construction of the time front was controlled by an optimization algorithm that guaranteed smooth connection of output contents in both temporal and spatial domains. By removing chronological constraints, this achieved smooth appearance of object motion in the final mosaic video. However, the loss of chronological ordering of object movements may lead to an incorrect representation of activities occurring in the scene.

2.3 Summary

The literature presented in Section 2.2.1 generates static panoramas based on parallax-free inputs. Even when combined with optimization methods, such as the work of Uyttendaele et al. [60], the ability to compensate for ghost errors due to parallax or object motion is still very limited. Manifold approaches, described in Section 2.2.2, combined with an optimization algorithm based on smoothness criterion, such as that implemented by Agarwala et al. [1], offer promising results for static mosaics. However, the requirement of a dense sampling of the scene is impractical

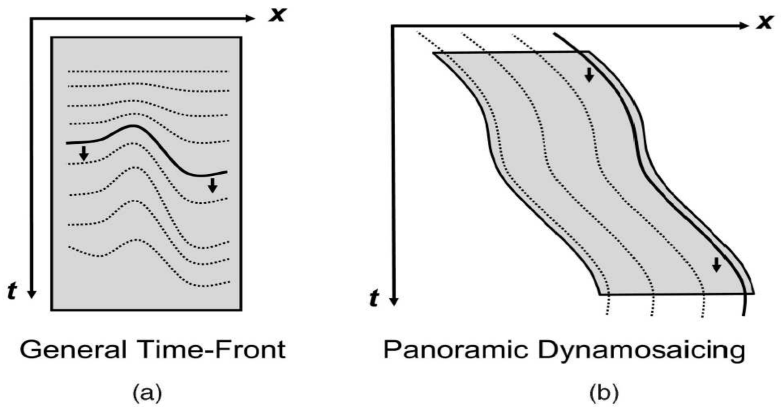


Figure 2–3: An illustration of synthesizing mosaic frames using time front to scan the space-time volume, in which t represents a time axis and x corresponds to the horizontal axis in image plane. In both (a) and (b), the time fronts across different input frames constitute snapshots in dynamic mosaic video thus, obviously, the chronological order of object motions are broken. Figure 6 in Rav-Acha et al. [4] (©[2007]IEEE, reproduced here with permission).

for many applications. State of the art dynamic mosaicing algorithms, introduced in Section 2.2.3, generally assume limited parallax resulting from camera motion, which too is often unrealistic. Moreover, none of the algorithms surveyed here can build a real mosaic video that preserves both motion consistency and chronological ordering of motion as appears in the input video.

In summary, the previous approaches all impose constraints of either a parallax-free camera configuration or a dense sampling of the scene, which, as discussed in Section 1.1, may not be sufficient to address the requirements of all applications. Based on our need for a true panoramic video of moving objects, these limitations motivate our effort to develop new techniques that can produce smooth mosaicing results, even in the case of a highly limited number of input sources and a scene with large depth variance.

CHAPTER 3
Image Mosaicing through Virtual Dense Sampling

The first solution to deal with the problems of parallax was inspired by manifold techniques that use dense sampling of the environment to build multi-perspective projection panoramas. As the general camera configuration, e.g. a small number of cameras in fixed positions along a large base line, usually cannot itself generate such a dense sampling, we turn to synthesis as an alternative. Starting from a pair of stereo cameras with a large baseline, along with their calibration parameters, a set of virtual images is synthesized from positions in between source cameras to compensate for the limited sampling rate. Next, an appearance-based optimization procedure is applied to select the set of strips from these real or virtual frames to build the final mosaics. Experiments indicate significantly improved mosaic results over competing methods.[1]

3.1 Synthesis of Virtual Images

The synthesis of virtual frames is performed using the plane sweep algorithm [16]. The basic idea is that when a set of parallel planes sample a $3D$ volume, the position within each plane, where rays received by different input cameras and containing the similar colors intersect, probably represents an object point of the scene. After

[1] The work in this chapter is based on the contents of the author's BMVC publication [42].

collecting the depth information of these potential object points, we project them to the image plane of a target virtual camera to build the synthesized output frame. The implementation of this idea includes the following three steps.

3.1.1 Building Intermediate Images through Plane Warping

Critical to the image mosaicing operation, the movement (both translation and rotation) between neighboring cameras must be smooth and continuous. Therefore, the virtual sources are positioned at equal intervals between the physical cameras and their orientations are interpolated smoothly between the rotation matrices of the source cameras.

Given the position and orientation of the virtual camera, we project pixels from input images onto sampling planes at different depths, and then re-project them onto the target virtual camera plane. Therefore we generate a set of intermediate images, each of which is actually the image of an xy sample plane at depth d and observed from the target virtual camera, as shown in Figure 3–1(d-f). In the plane sweep algorithm, this two-step process can be achieved through one warping operation performed by a projective mapping transformation known as a *homography.*

In general, the image $X_{image} = [x\ y\ 1]'$ of an object point $X_{world} = [X\ Y\ Z\ 1]'$ in the sampling plane at depth d and observed by a camera with the projection matrix $P = [p_1\ \ p_2\ \ p_3\ \ p_4]$ is given by

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ d \\ 1 \end{bmatrix} = [p_1\ \ p_2\ \ d \cdot p_3 + p_4] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = M_{3\times3}\bar{X} \tag{3.1}$$

where $M_{3\times3} = [p_1\ \ p_2\ \ d{\cdot}p_3{+}p_4]$, which only depends on the parameters of the camera projection matrix and the depth value of this sampling plane, and $\bar{X} = [X\ \ Y\ \ 1]'$. Thus the mapping between the projection of an object point $\bar{X}$ in a source image and that in an intermediate image is

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{intermediate} = M_{intermediate} M_{source}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{source} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{source} \tag{3.2}$$

where $H = M_{intermediate} M_{source}^{-1}$ is the homography that warps the source image into intermediate images corresponding to sampling plane at different depth levels d.

Each intermediate image contains RGB color channels, which are computed by a weighted pixel-wise averaging of warped source images. Every intermediate image also contains an associated matching score channel that measures the similarity between the projections from the input images. The matching score is calculated based on the aggregated sum of squared differences between these projections over supporting windows. We apply the approach of multi-resolution aggregation as introduced by Yang et al. [63] to achieve the robustness of similarity measurement in not only the textureless regions but also the regions across the depth boundaries. The pixels with low matching scores appear to be in focus in the intermediate images, as shown in Figure 3–1(d-f).

The number of intermediate images required is identical to the number of sampling planes, which should be located at the depth levels that best approximate the disparity distribution of the scene. In our experiments, these sampling planes are positioned at N evenly interpolated depth levels from Z_{min} to Z_{max}, where N is twice the maximal disparity value observed in the source images in order to accommodate the sub-resolution of disparities. Note that the X and Y axes of the world coordinate system define a plane that is parallel to all the sampling planes, while the Z axis points in the direction of depth.

3.1.2 Optimization through Graph Cut

The set of intermediate images are stacked together into a volume $V(x, y, z)$, where (x, y) represents pixel coordinates and z corresponds to the depth level. The one-parameter representation of $V(z)$ indicates an intermediate image at depth z. Generating a synthesized virtual image is actually a procedure that maps pixels from the desired output image to their correspondences in intermediate images. This mapping only takes place in the depth domain provided that the matching pixels do not have position shift within the image plane, so that the output pixel $p(i, j)$ may correspond to those at the same position (i, j) in intermediate images. Let P denote the set of pixels in the output virtual image and L be the set of depth levels $\{d_1, \cdots, d_N\}$. The problem of building a synthesized image is defined as:

Problem. *Given N intermediate images, find a labelling $f : P \rightarrow L$ that assigns each output pixel a proper depth level d_i, $i \in [1, N]$, so that the energy of the labelling f is minimized.*

The energy of the labelling is defined as:

$$E(f) = E_{data} + E_{smoothness} \tag{3.3}$$

The first term, E_{data}, measures the cost of assigning the set of depth labels to pixels in the virtual image. In our application, E_{data} is related to the matching cost of the entire synthesized virtual image as:

$$E_{data} = \sum_{p \in I_{output}} A(p) \tag{3.4}$$

where I_{output} is the synthesized virtual image and $A(p)$ is defined as:

$$A(p) = \max(\phi(p, d(p)), \tau) \tag{3.5}$$

where $\phi(p, d(p))$ is the matching cost of pixel p in the intermediate image $V(d(p))$, if p is assigned the depth value, $d(p)$. Here τ is a constant, which is chosen empirically based on the color distribution of the input images. A lower matching score represents a higher probability that the pixel corresponds to an object located on the sampling plane at depth $d(p)$.

The second term, $E_{smoothness}$, measures the cost of assigning depth values to a pair of neighboring pixels and is used to indicate the smoothness of the depth transition from one pixel to its neighbors, i.e., neighboring pixels having similar colors should typically have similar depths as well:

$$E_{smoothness} = \sum_{p \in I_{output}} \Big(\sum_{\{q \in N_p | d(p) \neq d(q)\}} B(p, q)\Big) \tag{3.6}$$

where (p, q) is a pair of neighboring pixels with different depth values. N_p is a neighboring system around pixel p, such that $|x_p - x_q| + |y_p - y_q| = 1$. $B(p, q)$ is an increasing function of $\Delta I(p, q)$, the abstract color difference between the pair of pixels (p, q), and defined following the work of Boykov et al. [9] as:

$$B(p,q) = \begin{cases} 3\lambda & \text{if } \Delta I(p,q) < 5 \\ \lambda & \text{otherwise} \end{cases} \tag{3.7}$$

where λ is a value chosen empirically based on the color distribution of input images. $B(p, q)$ generates a higher penalty value, 3λ, when two neighboring pixels happen to have similar colors ($\Delta I(p, q) < 5$), but different depth values. Although the advanced quadratic $E_{smoothness}$ term maybe be considered as an alternative, our experiments suggest that the $B(p, q)$ defined by Boykov et al. [9] works sufficiently well.

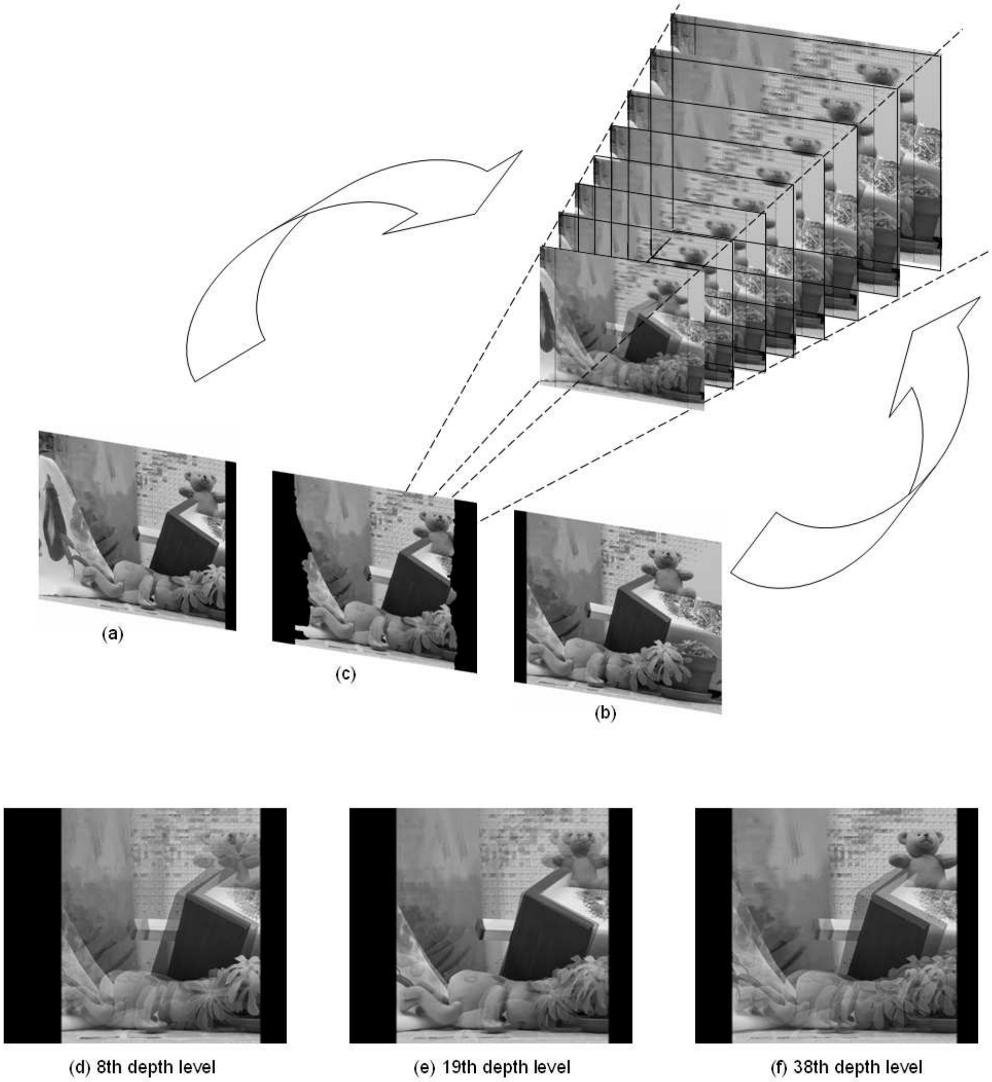
With the energy definition of Equation 3.3, an alpha-expansion *graph cut* algorithm[2] [8][9] is applied to find the optimal labelling f.

3.1.3 Synthesizing Virtual Images

Given the optimal labelling $f : P \to L$, for a certain pixel p in the output image, the color of its corresponding pixel in the intermediate image at depth $f(p)$ is pasted into the final synthesized virtual image. In this manner, an output is built as illustrated in Figure 3–1c.

When two input cameras are located along a wide baseline, as is the case in our test data, there are likely to be some portions of the 3D scene that are visible by one camera but occluded in the other. Pixels in these regions have matching

[2] Many computer vision problems, e.g., the stereo correspondence problem, can be formulated in terms of energy minimization. The corresponding minimum energy solution is calculated as a minimal cut of the graph by the max-flow min-cut theorem. Such an algorithm is called a graph cut.



(d) 8th depth level

(e) 19th depth level

(f) 38th depth level

Figure 3–1: Illustration of the plane sweep algorithm. Given two input images (a) and (b), we first warp them onto the parallel sampling planes, stacked at different depth levels along the z axis. Comparing the images of these sampling planes, i.e., the intermediate images, we synthesize a virtual image (c) at a position between the inputs. Examples of intermediate images are illustrated in (d)(e)(f). Note that different objects in the scene are in focus at the different depth levels, where their projections from different inputs coincide well with their real $3D$ positions.

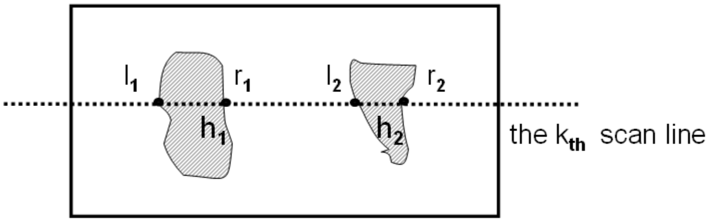


Figure 3–2: There are two shaded hole regions in this figure. On the k_{th} scan line, points l_1, r_1 and l_2, r_2 are the pairs of non-hole-region depth values on both sides of the segments across hole regions h_1 and h_2 respectively. We fill segments on the kth scan line in the hole region h_1 with the higher depth value, i.e. the one further away from the camera, of l_1 and r_1 and do likewise to fill the segment of hole h_2.

scores, $\phi(p, d(p))$ as appeared in Equation 3.5, beyond the threshold, and thus all receive a special depth label as "Occluded" during the graph cut optimization. These pixels constitute hole regions appearing in the virtual images and must be filled using depth information from neighboring pixels. To do so, we scan the hole region row by row. As illustrated in Figure 3–2, for any row, each hole h_i can be delineated on both sides by the nearest non-hole points, l_i and r_i, with corresponding depths determined by the plane sweep operation. Assuming that a distant hole is occluded by objects closer to the input cameras, we estimate its depth by choosing between the two points l_i and r_i the one further from the camera. Once all the depth values for the hole regions are estimated in this manner, we can use a forward mapping to project the corresponding input images, i.e., if l_i is chosen then we use the left source and, conversely, if r_i is chosen, we use the right source, onto the virtual image plane at the specified depth, thereby filling in the missing texture of the virtual images. Although more advanced hole-filling algorithms, such as inpainting, are well known, our experiments suggest that when, as in our cases, inputs contain translationally dominant camera motion or are rectified, the simple strategy described above works sufficiently well for the small holes typically observed in the virtual images.

Determining the number of virtual images needed is a difficult sampling analysis problem [51]. Obviously, there is a minimal number of virtual images, which is necessary to guarantee the anti-aliased rendering of the final mosaic result. This minimal sampling rate has a complex relationship with the depth range, texture complexity, and resolution of input images. It is defined according to the boundaries of the depth range regardless of the complexity of depth variance in the scene. A big difference between Z_{min} and Z_{max} may lead to a higher minimal sampling rate. If the texture of the scene presents less variation and the input images are of lower resolution, a smaller minimal number of virtual images may be sufficient. We generate 50 virtual images, which are sufficient for any of the data sets we tried in our tests.

3.2 Generating the Mosaic through Appearance-Based Optimization

A good mosaic result should exhibit minimal visual or perceptual artifacts, i.e., each local region in the panorama should resemble its corresponding area in the source images. Assuming source images all present smooth appearances, the constructed panorama should ideally inherit this smoothness, especially in the regions where patches from different sources are merged. This smooth appearance transition has become an important criterion to judge the quality of mosaic results [1][3][4]. Thus, in this section, we introduce a technique that uses appearance, i.e., texture, optimization, to build the final panorama from a dense sampling, which now includes both real and synthesized images of the environment. Similar to other work [62], we generate the final mosaic by combining thin vertical strips from source images along a minimum-cost path calculated by *dynamic programming* [19], based on the smooth connection criterion.

Compared to traditional manifold mosaicing algorithms, the new technique does not explicitly estimate the geometrical relationship between input frames. Instead, the analysis of texture similarity among strips in the search space automatically determines the group of slices that should be connected side by side in the mosaicing plane. Furthermore, unlike traditional algorithms, which only consider several successive frames in their image alignment, the new technique determines a stitching path that considers the matching performance across the sequence of entire dense sampling frames. This may provide a mosaic result closer to the global optimum solution than any conventional manifold mosaicing algorithm.

3.2.1 Problem Definition

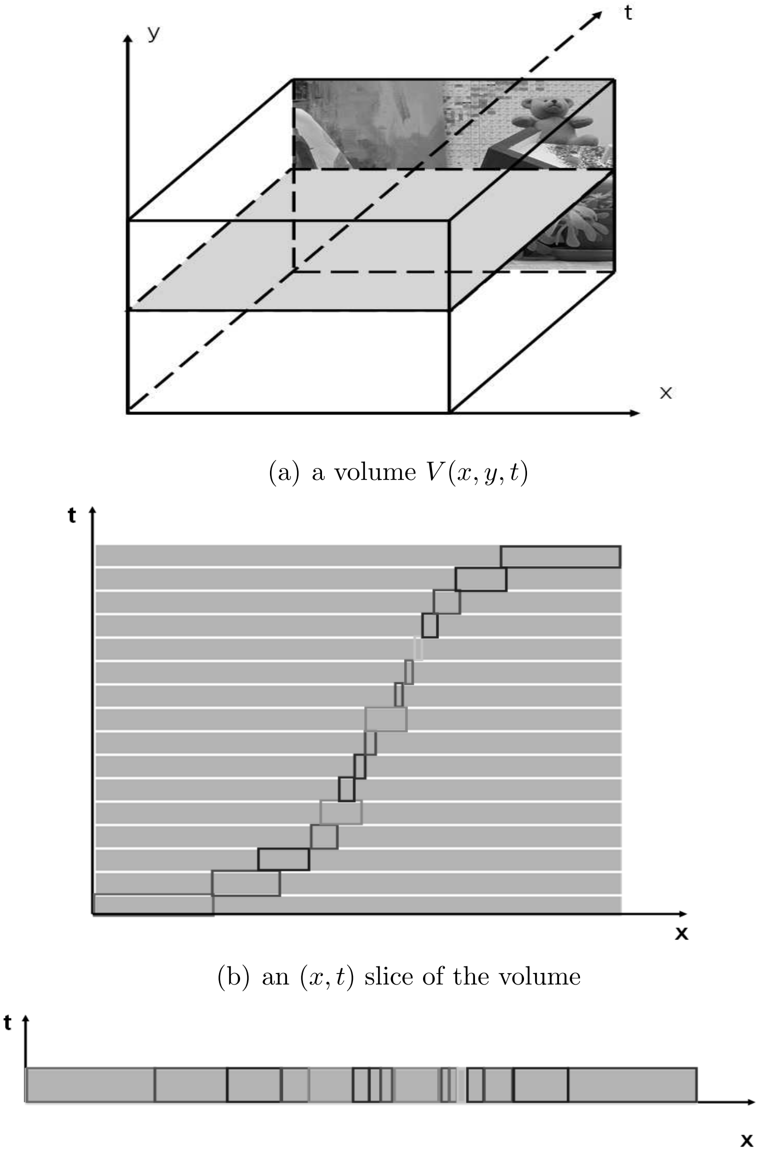
As explained above, the virtual images in this dense sampling set are synthesized at discrete camera positions along a continuous path between two input cameras. They can thus be regarded as a sequence of video frames taken by a camera moving along this path. These frames form a spatiotemporal volume $V(x, y, t)$, where x, y parameterizes the space as pixel coordinates, and t parameterizes time as the index of the camera position along the acquisition path. The two coordinate representation of $V(x, t)$ means the x'th column taken from the t'th frame. In an (x, t) slice (one scanline through time) of a sample input volume, as shown in Figure 3–3(a-b), each input image is represented as a horizontal gray bar. The group of column strips are first selected from inputs according to the smoothness connection criterion, and then stitched side by side to build the output mosaic, which is of larger size than any of the input images, as shown in Figure 3–3c.

As suggested by Peleg *et al.* [40], in our case of a translationally dominant camera motion model, it is reasonable to choose vertical strips from input images to generate the mosaic results. Thus, columns in input images become the fundamental element in the algorithm. Note that the rectification is not strictly necessary in this approach. As long as input images do not contain changes of focal length and the effects of forward or backward motion are significantly smaller than other camera motion in the inputs, the new algorithm may build reasonable mosaic results by copying columns from the virtual dense sampling set.

Building a final mosaic result is represented as a mapping Γ from columns in the output mosaic to columns in input images. The mapping $\Gamma(\theta_i)$ is a vector of the form $(x, \Delta y, t)$, which maps any output column θ_i to an input column $V(x, \Delta y, t)$, in which the Δy represents a vertical shift value for the column $V(x, t)$. For most of our test cases, Δy equals zero, either because we use rectified inputs or the vertical motion between cameras is trivial.

The problem of creating a final mosaic is defined precisely as follows:

Problem. *Given the set of virtual dense samples as obtained by a camera with translationally dominant movement along a continuous path, we need to find the mapping* $\Gamma(\theta)$ *for every column in the output panorama, such that the connection cost of the panorama is minimized.*



(a) a volume $V(x, y, t)$

(b) an (x, t) slice of the volume

(c) the output panorama

Figure 3–3: (a) The spatiotemporal volume $V(x, y, t)$. (b) An (x, t) slice of the volume, as indicated by the gray plane in (a). Each sampled image is represented as a horizontal gray bar in an (x, t) slice. The group of column strips are first selected from inputs according to the smoothness connection criterion, and then stitched side by side to build the final mosaic. (c) The diagram shows the output panorama, which is larger than any of the input images.

Assuming the final mosaic contains M columns, $\{\theta_i | i = [1, M]\}$, the connection cost of this mosaic result is defined as:

$$\text{Cost}(\Gamma) = \sum_{i=1}^{M} C_s(\Gamma, \theta_i) \tag{3.8}$$

where

$$C_s(\Gamma, \theta_i) \quad = \|\Gamma(\theta_i) - \Gamma(\theta_{i+1})\|^\gamma \tag{3.9}$$

Here, γ is a constant, used as an exponent on the L_2 norm. If $\Gamma(\theta_i) = V(j, 0, k)$ and $\Gamma(\theta_{i+1}) = V(g, 0, h)$, which means the columns θ_i and θ_{i+1} in the output panorama correspond respectively to the columns $V(j, k)$, the j'th column taken from the k'th input sampling frame, and $V(g, h)$, the g'th column taken from the h'th input sampling frame, then Equation 3.9 becomes

$$C_s(\Gamma, \theta_i) \quad = \|V(j, k) - V(g - 1, h)\|^\gamma \tag{3.10}$$

The total connection cost of the panorama is a summation of C_s over all the neighboring columns in the output. C_s is defined based on the smooth connection criterion. When column $V(g, h)$ is next to column $V(j, k)$ in the final mosaic, a good C_s value indicates high similarity between $V(j, k)$ and $V(g - 1, h)$, and thus, the transition from $V(j, k)$ to $V(g, h)$ appears as smooth as the local transition from $V(g - 1, h)$ to $V(g, h)$ in the same frame taken at the h'th position along the acquisition path.

3.2.2 Implementation

Finding the optimal mapping $\{\Gamma(\theta_i)\}_{i=1}^{M}$ in the spatiotemporal volume $V(x, y, t)$ to minimize the connection cost of the output panorama is an optimization problem. Given the starting point as the first column in the initial frame and the ending point as the last column in the final frame, a standard dynamic programming algorithm is used to calculate the shortest path, i.e., the path with the minimal connection cost, between these two terminals. Columns along the shortest path are then copied into the final mosaicing plane one by one and merged to build the output panorama.

Since pixel-wise intensity difference is unreliable in the analysis of stereo disparity [47], the connection cost between successive columns in the output panorama, as in Equation 3.10, should apply a window-based aggregation, i.e., an average filter, rather than column-wise comparison. $\|\Gamma(\theta_i) - \Gamma(\theta_{i+1})\|$ is calculated as the aggregated L_2 norm distance of column differences between two rectangular windows with centers at columns $\Gamma(\theta_i)$ and $\Gamma(\theta_{i+1})$ respectively.

Theoretically, the dynamic programming algorithm searches for the successor of column $\Gamma(\theta_i)$ among any column in volume $V(x, y, t)$ that has not been included in the path up to $\Gamma(\theta_i)$. Practically, however, it is not necessary to consider all these potential options, some of which are obviously very different from the current column. Instead, only those enclosed by a neighboring window in the spatiotemporal volume are compared, where the window size depends on motion velocity between input frames. Such constraints on the search space greatly reduce the computational cost, although at the risk of missing the globally optimal solution. However, it is the local smoothness, i.e., the smoothness of connections between neighboring columns, that is the primary determinant of quality of the output mosaic result. Thus, the local optimal solution obtained by a constrained dynamic programming search still produces a satisfactory panorama result. This will be illustrated in the next section of experimental results.

3.3 Experimental Results

To evaluate the quality of the proposed algorithm, we compare the results to those of traditional image mosaicing techniques, which typically generate excellent mosaics from source images constrained by the conditions outlined in Section 1.1. This comparison highlights the capacity of our approach to generate significantly improved mosaics from source images exhibiting non-trivial parallax. Although the significant degree of overlapping content between neighboring input images, required by traditional image mosaicing algorithms, is not necessary for our approach, some overlap is still necessary. We have not characterized this minimum quantitatively, as the value is likely to depend on image content. For our experiments, the input images exhibited an overlap over a minimum of 1/3 of their areas.

We use three data sets in the tests. Both the Middlebury teddy data set [48] and the data from Seitz [49] contain structured indoor or outdoor scenes, which exhibit a complex, wide range of depth distribution. The sample results provided here use as input two fixed video cameras with a large baseline and fixed parameters. Obviously, these results can be extended in a straightforward manner to take advantage of additional cameras. Camera calibration is performed using Zhang's [66] method, as implemented by Bouguet [7] for our own data, which will be used for the experiments in the next two chapters, and by the *structure-from-motion* algorithm [23] for externally supplied data.[3]

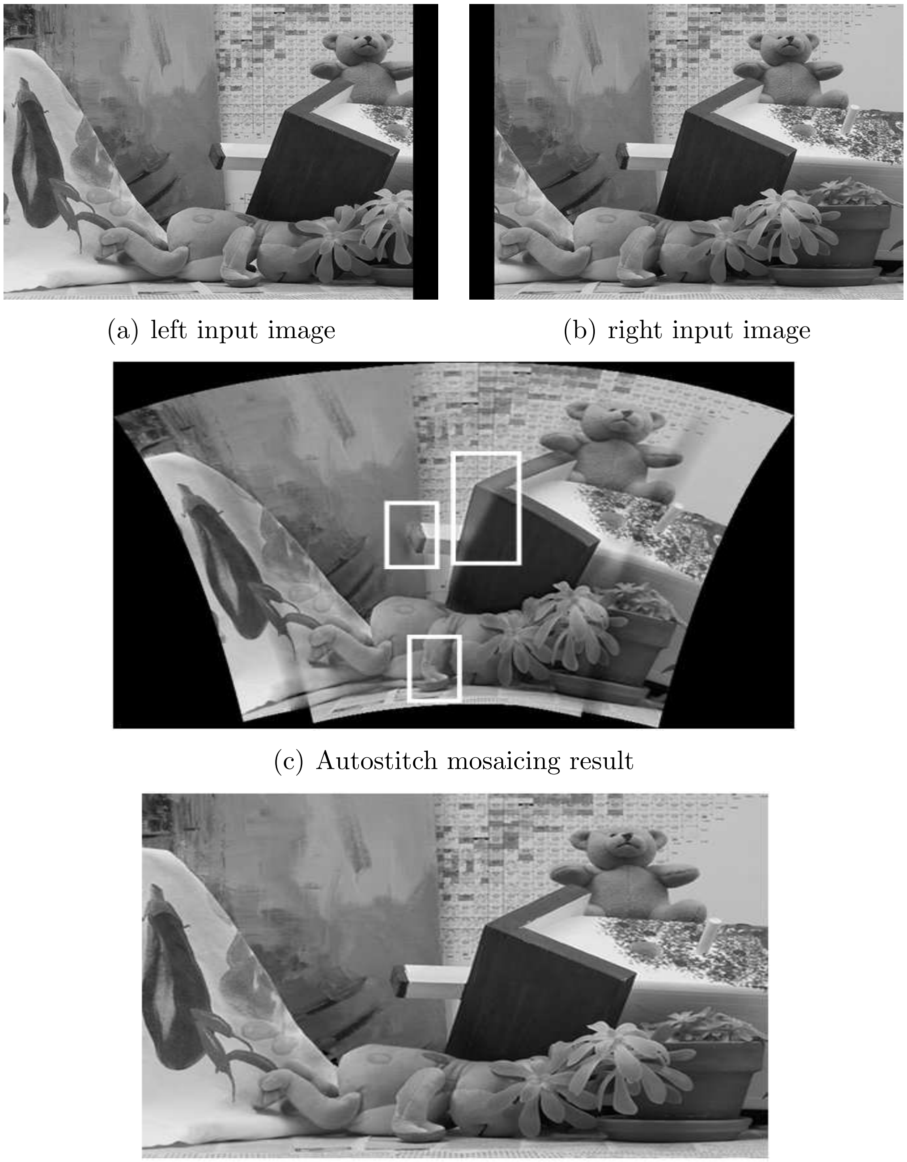
3.3.1 Comparison of Results with Autostitch

Autostitch [12], a state of the art representative of traditional image mosaicing techniques, may produce unpleasant artifacts in its mosaic results if given as input two sparse samples of the scene with complex depth variation.

The first kind of artifact, *distortion*, is mainly caused by disparity variance between objects at different depths from the camera. Autostitch has to deform the input images, compressing close objects and expanding distant objects to equalize their respective disparities. The normalized amount of overlap between the images allows for a smooth combination of the two deformed inputs. The results of Autostitch, seen in Figures 3–4, 3–5, and 3–6, contain a foreground region at the bottom of the mosaic, which has shrunk relative to the background at the top.

This stretching effect is proportional to the amount of disparity variance in the input images. For example, among the three data sets, the “house”, displayed in Figure 3–6, presents the smallest range between the maximal and minimal disparity values as the scene objects are all distant from the camera. Thus the Autostitch result of this data set exhibits the least distortion of compression or expansion.

[3] While the calibration was likely of sufficient quality so that the effects of calibration errors were not observed in the results, this could be a factor in general.



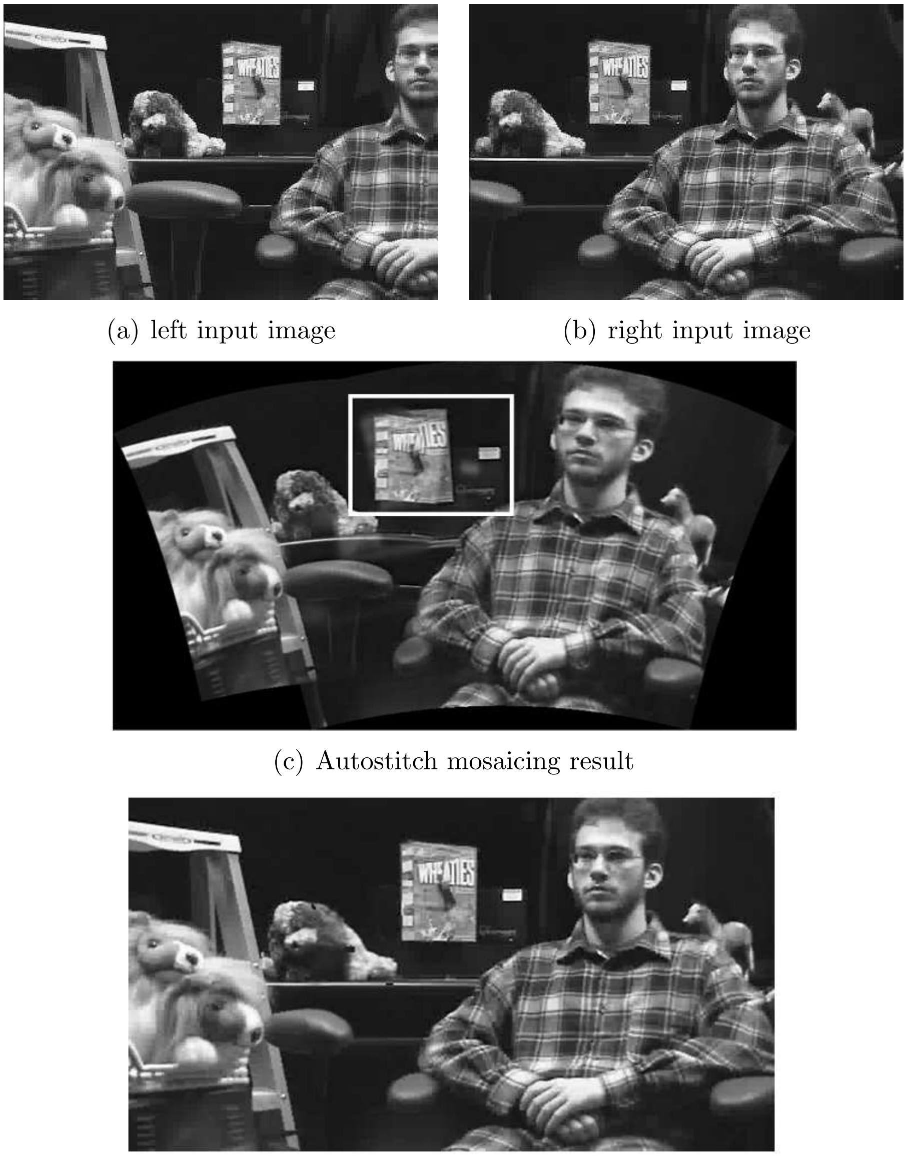
(a) left input image

(b) right input image

(c) Autostitch mosaicing result

(d) stitching result of our algorithm

Figure 3–4: Given two input images from the teddy data set, the mosaic results of Autostitch are compared to that of our algorithm. Regions with ghost errors are indicated by highlighted rectangles. These errors are caused mainly by inaccurate image alignment.



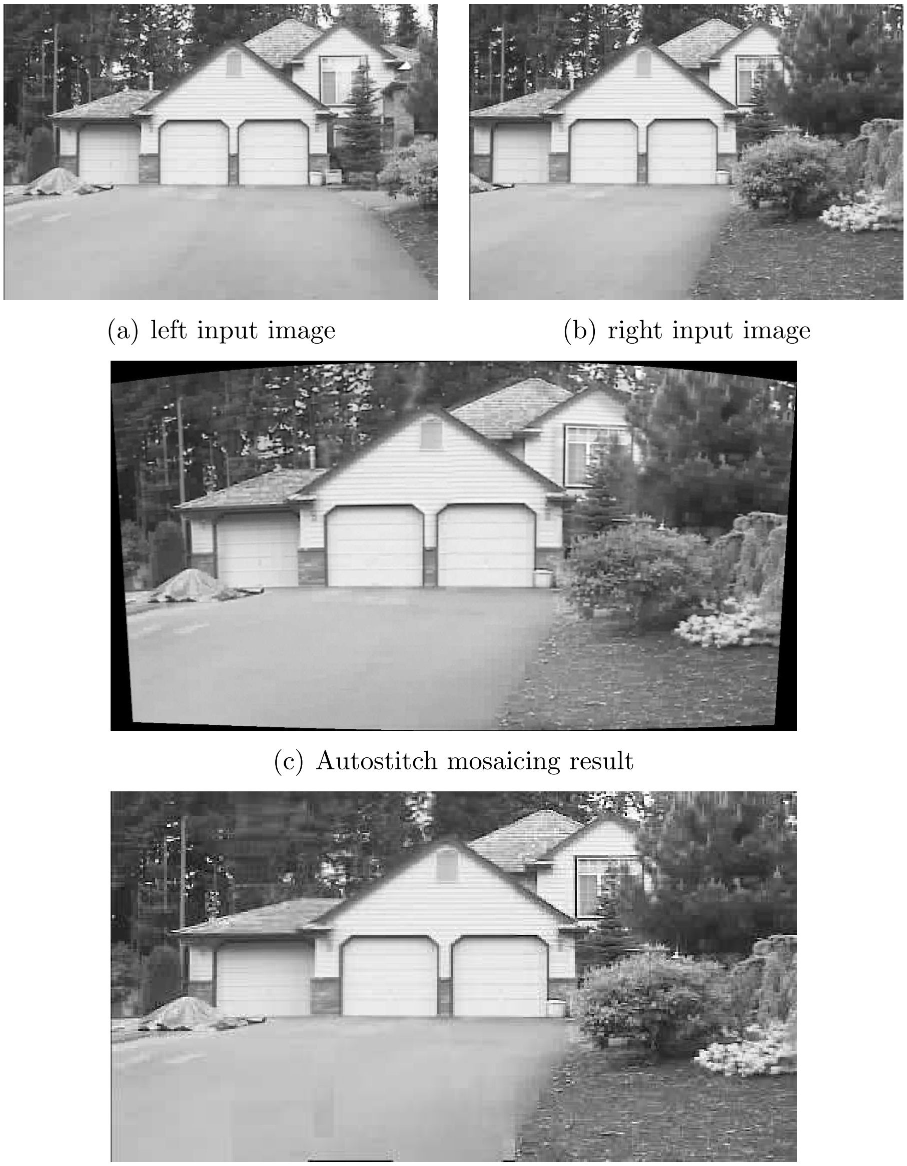
(a) left input image

(b) right input image

(c) Autostitch mosaicing result

(d) stitching result of our algorithm

Figure 3–5: Mosaicing results of Autostitch compared to our algorithm for the data set from Seitz. An example ghost error is indicated by a highlighted rectangle in (c).



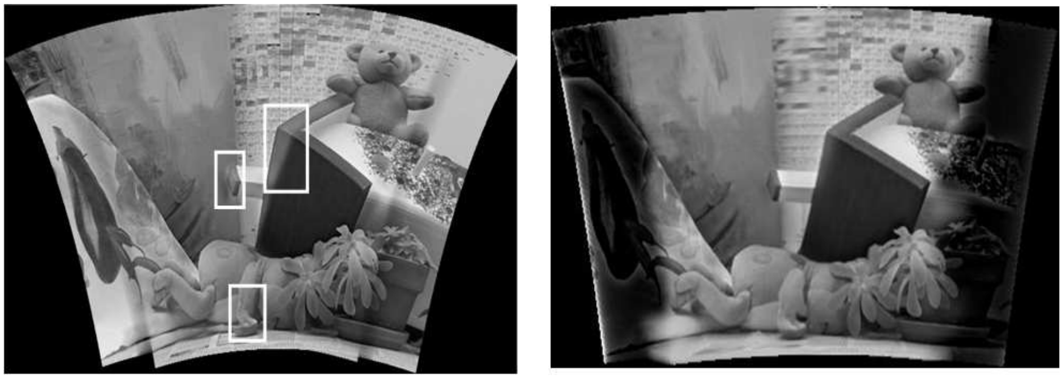
(a) left input image

(b) right input image

(c) Autostitch mosaicing result

(d) stitching result of our algorithm

Figure 3–6: Mosaicing results of Autostitch compared to our algorithm for the "house" data set from Seitz.



(a) mosaicing result using two real input images (b) mosaicing result using virtual dense sampling inputs

Figure 3–7: Compare the mosaic results of Autostitch using respectively the sparse real input images or the virtual dense sampling inputs. The ghost errors, in figure (a) and enclosed by the highlighted rectangles, are compensated by using the virtual dense samples, in return, the intensities and the resolution of the mosaic result, in figure(b), decrease in certain regions.

The second kind of artifact, *ghost error*, appears in the highlighted rectangular boxes in the Autostitch mosaic results of Figure 3–4 and 3–5. As observed in these results, accurate image alignment remains difficult even after compensating for disparity differences by compressing the foreground and expanding the background objects. This is due to the depth variance in the scene and the significant view disagreement between two cameras with a wide baseline.

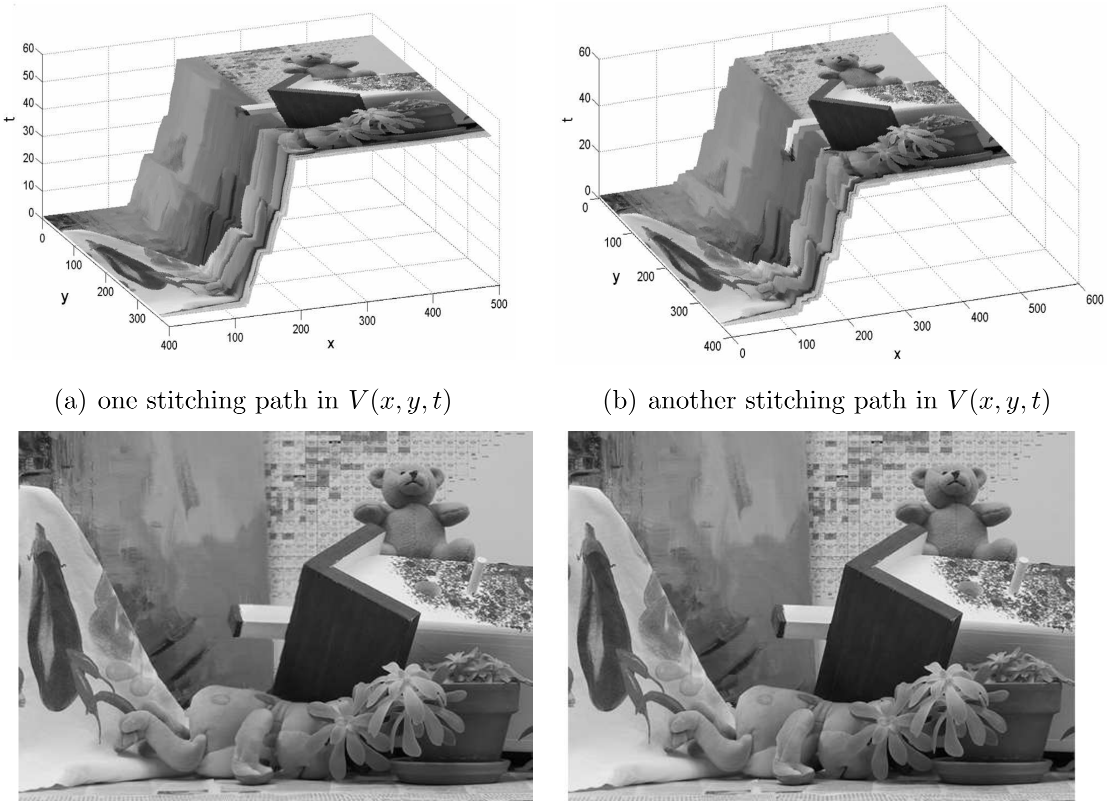
In contrast, the new algorithm generates results free of such artifacts, and thus satisfies the desired properties summarized in Section 1.1. As is evident, the parallax effects have been greatly reduced by the incorporation of synthesized virtual images. This is illustrated in the results of Figures 3–4, 3–5, and 3–6.

In theory, the use of virtual dense samples should also benefit Autostitch to eliminate the ghost artifacts due to image alignment errors and reduce distortion of compression and expansion. As illustrated in Figure 3–7, the ghost errors enclosed by the highlighted rectangles in the mosaic, resulting from sparse inputs, are visibly reduced by the use of virtual dense samples. However, the blending of 52 highly overlapped inputs, as shown in Figure 3–7b, not only degrades the intensity level on both sides of the mosaic, i.e., the non-overlapping regions of the real inputs, but also decreases the resolution of the mosaic result, as is particularly evident in the blurred chart on the background wall. These problems do not occur with our algorithm in its use of the same dense samples.

3.3.2 Locality of Smooth Connection in Mosaicing Results

Determining the quality of a particular image mosaic is largely based on the criterion of smoothness and continuity of its appearance. In our algorithm, any mosaic result built on the path calculated by dynamic programming satisfies this criterion.

We manually choose a reference node in $V(x, y, t)$, which splits the stitching path into two segments. Different choices of this reference node lead to different paths through the volume. Comparing the two results in Figure 3–8, which have reference nodes ($x = 130,\ t = 2$) and ($x = 162,\ t = 11$) respectively, we find that they both successfully combine the input information into the panorama output without inducing artifacts of missing or duplication of objects. Perceptually, it is difficult to judge which of the two is of better quality than the other, as there are multiple paths through the spatiotemporal volume $V(x, y, t)$ that all provide reasonable outputs.



(a) one stitching path in $V(x, y, t)$ (b) another stitching path in $V(x, y, t)$

(c) mosaic result of path (a) (d) mosaic result of path (b)

Figure 3–8: Comparison of mosaic results from two stitching paths. Path(a) goes through the node ($x = 130,\ t = 2$) and contributes to the mosaic result of size 499 × 375, while path(b) passes the node ($x = 162,\ t = 11$) with a corresponding mosaic result of size 506 × 375. Despite their differences, both mosaic results are perceptually correct.

As noted earlier in Section 3.2.2, constrained dynamic programming may not find the global minimal cost stitching path although it achieves more efficient computational performance. Nevertheless, the mosaic result from a locally optimal stitching path is still perceptually satisfactory.

3.3.3 Computational Expense Analysis

The analysis of computational complexity of the virtual dense sampling image mosaicing algorithm (VDS) is a challenge, since the VDS contains a sequence of processing steps of camera calibration, the synthesis of virtual dense samples, and finally, the generation of mosaics. Most significantly, the computational cost is related to the input parameters, such as the resolution of input frames, and the number of depth levels assumed. We conducted a run-time analysis for the step of synthesis of virtual dense samples, varying both of these parameters. As shown in Table 3–1, computational cost increases exponentially with the number of depth levels and (approximately) with the square of the resolution of the input images.

3.3.4 Limitations

With the help of virtual dense sampling, our algorithm can construct reasonable mosaics given inputs taken under most camera motion patterns, such as the pure translation of the Teddy data set, and the dominant translation with a panning rotation in the data from Seitz. However, because the algorithm has not yet considered the scaling factor, changes of scale or resolution between input images, caused by zoom adjustment or forward/backward camera motion, are beyond the capability of our algorithm.

Table 3–1: Comparison of run-time (in seconds) under different combinations of input parameters for the synthesis of virtual frames. Table (a) illustrates the exponential relationship between number of depth levels and computational cost while Table (b) illustrates the approximately square relationship between input resolution (in pixels) and computational complexity. The second row in Table (b) provides a normalized resolution value, based on the scale of 225 × 187 pixels as equivalent to 1.0.

(a)

depth levels	11	22	33	44	55	66	77
run-time (s)	0.60	1.77	4.43	8.13	13.14	22.03	31.64

(b)

input resolution	225 × 187	270 × 225	315 × 262	360 × 300	405 × 337
normalized resolution	1	1.44	1.96	2.56	3.24
run-time (s)	13.14	36.10	76.94	174.67	247.34

The algorithm performs well for static scenes, although does not usually generate consistent mosaic results over time if there are dynamic objects in the environment. Object motion induces texture changes in images, which result in varying stitching cost for paths in the volume $V(x, y, t)$, and in turn, lead to different optimal stitching paths between successive frames. If used to generate a mosaic video, this will typically result in discontinuities of object movement.

Moreover, for each output mosaic frame, the regeneration of the set of synthesized virtual images and the recalculation of the optimal stitching path represents a high computational expense. When computation time is a serious consideration, our algorithm is obviously not an ideal choice.

3.4 Summary

This chapter poses the problem that traditional mosaicing techniques typically fail to generate reasonable mosaic results when given a limited number of inputs with non-trivial parallax.

Our novel image mosaicing algorithm built on the techniques of image-based rendering and optimization methods offers a potential solution to this problem. Experiment results demonstrate that the new method outperforms conventional image mosaicing techniques when presented with challenging inputs.

However, the high computational requirements and inability to handle dynamic objects prevents the use of this technique to achieve more general goals, such as construction of dynamic mosaic video.

CHAPTER 4
Depth Based Image Mosaicing

To overcome the parallax issues in image mosaicing, this chapter introduces a new method that uses a camera projection procedure, with depth estimates generated from a virtual mosaicing camera, to construct the final panorama. Experiments indicate the capability of the new approach to generate improved results over traditional algorithms when given challenging sources, such as sparse input samples that exhibit non-trivial parallax effects. Furthermore, the ability of the new approach to deal with more diverse environments, e.g., containing dynamic objects, at a lower computational cost reflects its superiority over the algorithm presented in the previous chapter.[1]

4.1 Introduction

Traditional techniques construct mosaics by first aligning, i.e., registering, the input images, and then warping and stitching them in mosaicing surfaces. However, when there is considerable disparity variance between foreground and background contents, the general assumption of a planar scene is no longer applicable. In this case, the image alignment procedure applied by traditional image mosaicing algorithms may fail to determine a common motion model that is appropriate for both

[1] The work in this chapter is based on the contents of the author's ICPR publication [43].

near and far objects in the scene. Even if it manages to estimate such a transformation model, the resulting distortion of image contents leads to perceptually unacceptable mosaic results.

In summary, state of the art image mosaicing algorithms are incapable of handling the effects of parallax in inputs due to a complex depth distribution in the scene. However, assuming a depth map of the entire environment is available, we may nevertheless synthesize the panorama as if seen through a virtual camera with a wider FOV. Provided that the depth estimates are reasonable, we may build a panorama free of parallax-related artifacts, e.g., distortion, duplication or missing objects.

In the output mosaic result, pixels corresponding to objects in overlapping regions, R_o, which are observed by multiple input cameras, are synthesized using traditional image-based rendering techniques. Pixels from non-overlapping regions, R_{non}, and consequently, lacking stereo information, are neglected by traditional view synthesis techniques. As an improvement, the new approach described in this chapter provides a solution to include such pixels from R_{non} into the mosaic result.

The remainder of this chapter is organized as follows. The synthesis of mosaics in overlapping and non-overlapping regions are discussed in Section 4.2 and Section 4.3 respectively. Section 4.4 provides a comparison of experimental results with those of Autostitch and those using ground truth depth values. Section 4.5 discusses the limitations of the proposed algorithm. Finally, a brief summary is provided in Section 4.6.

4.2 Mosaic Synthesis in Overlapping Regions

The synthesis of mosaics in overlapping regions, R_o, is performed by the plane sweep algorithm [16] using the method introduced in Section 3.1.

The parameters of the virtual camera that is used to build the output panorama are chosen as the smooth interpolations between those of input cameras. However, because the output mosaic has larger image size than any of the sources, the corresponding elements in the internal parameter matrix of the virtual camera must be adjusted accordingly. Without loss of generality, in our test, we only consider the enlargement of the image size along the horizontal, x, direction.

Given the selected pose and the internal parameters of the virtual camera, the inputs are then projected onto parallel planes located at different depths to generate a set of intermediate images, $\{I_{d_i}\}_{i=1}^{N}$, as illustrated in Figure 3–1 (d-f). Each intermediate image represents a sampling plane that discretizes the $3D$ space at depth level d_i.

Let the labelling $f : P \rightarrow L$ be a procedure assigning every output pixel, $p \in R_o$, a proper depth level in the set $L = \{d_i\}_{i=1}^{N}$. With the optimal solution of this labelling calculated by a graph cut algorithm [8][9], each pixel, $p \in R_o$, finds its correspondence in the intermediate image $I_{f(p)}$. Pasting the color of the corresponding pixel for every $p \in R_o$ into the final output, we construct a mosaic result exhibiting contents only in R_o, as seen in the regions enclosed by black bounding boxes in Figure 4–4 and Figure 4–6. Construction of the mosaic result for the non-overlapping regions will be introduced in the next section.

4.3 Mosaic Synthesis in Non-Overlapping Regions

The areas in source images that correspond to the FOV of only a single input camera contribute to R_{non} of a mosaic result by projecting them onto the mosaicing image plane according to their depth estimates. However, as these are observed only by a single input camera, their depth estimates must be calculated by a different method from that applied to R_o of the mosaic, which can make use of stereo correspondences. Such a method is the topic of this section.

We observe that depth discontinuities rarely occur in regions of uniform texture but typically coincide with color segment boundaries [57][65]. Thus, taking advantage of color segmentation in the sources, (reliable) depth information of color segments in R_o can be propagated to adjacent color segments in R_{non}, provided that this results in an appearance of smooth connection between them.

4.3.1 Color Segmentation

Input images are first decomposed into color segments using the mean-shift-based segmentation algorithm that incorporates edge information as proposed by Georgescu *et al.* [21]. These color segments are then divided into two groups. One group, which already has reliable depth estimates, contributes to R_o in the mosaic result. The second group, which lacks depth estimates, constructs R_{non}. An example segmentation result of the data from Seitz is shown in Figure 4–1(c-d), where hashed color segments represent the R_{non} in the output mosaic.

Generally, as long as boundaries along color discontinuities are preserved, a color segmentation with larger segments is preferred to an over-segmentation. Minimizing the number of color segments reduces the computation in the depth propagation procedure, the details of which will be addressed in the following sections. Furthermore, over-segmentation into small size segments may result in poor depth estimates in textureless regions.

4.3.2 Problem Definition

Because of the lack of available $3D$ information to establish the shape functions for color segments in non-overlapping regions, we naively assume that each color segment in R_{non} is of uniform depth. While this may not be true in practice, our experiments suggest that the mosaic results based on this assumption of uniform depth are, nevertheless, still perceptually correct.

Given the set of intermediate images, $\{I_{d_i}\}_{i=1}^{N}$, built when synthesizing the mosaic in overlapping regions, the construction of mosaics in non-overlapping regions involves mapping the color segments from R_{non} of the output mosaic to their correspondences in intermediate images. Let S denote the set of color segments $\{s_1, s_2, \ldots, s_M\}$ in R_{non} and L be the set of depth levels $\{d_1, \cdots, d_N\}$. Based on the assumption of uniform depth, the problem of synthesizing the non-overlapping regions in the output mosaic is defined precisely as follows:

Problem. *Given N intermediate images, find a labelling $\rho : S \to L$ assigning each color segment in the non-overlapping regions of the output mosaic a proper depth level d_i, so that the energy of the labelling ρ is minimized.*

The energy of the labelling is defined as:

$$E(\rho) = E_{smoothness} + E_{occlusion} \tag{4.1}$$

The first term, $E_{smoothness}$, evaluates the overall connection cost between neighboring color segments as follows:

$$E_{smoothness} = \sum_{i=1}^{M} \sum_{(p,q)\in\Psi} C_i(p,q) \qquad (4.2)$$

where $\sum_{(p,q)\in\Psi} C_i(p,q)$, with respect to one color segment s_i ($s_i \in S$), equals the total connection cost of all pairs of neighboring pixels, (p,q), within the region Ψ, which represents border areas between the color segment s_i and its neighbors that already have depth estimates.

In Equation 4.2, the connection cost of one pair of neighboring pixels is calculated as follows:

$$C(p,q) = |I_{D(p)}(p) - I_{D(q)}(p)|^2 + |I_{D(p)}(q) - I_{D(q)}(q)|^2 \qquad (4.3)$$

where $D(p)$ and $D(q)$ are the depths of p and q respectively. $I_{D(p)}$ refers to the intermediate image at the depth $D(p)$ and $I_{D(p)}(p)$ represents the intensity of pixel p in this intermediate image. The transition between p and q is smooth when the local region containing this pair of pixels in the output mosaic resembles the corresponding areas in the intermediate images. With such a definition, $C(p,q)$ is minimized when the two patches in $I_{D(p)}$ and $I_{D(q)}$ respectively, and both containing the pair of neighboring pixels as (p,q), exhibit similar texture around (p,q).

If at every possible depth level, the smooth connection cost, $\sum_{(p,q)\in\Psi} C_i(p,q)$, of color segment s_i exceeds a threshold, or the number of neighboring pixel pairs in its Ψ region is insufficient to generate a reliable depth estimate, s_i is defined to be *occluded*. Accordingly, its smooth connection cost in Equation 4.2 is set to zero.

In order to ensure that the energy function 4.1 is not biased towards a trivial optimum by considering all color segments as occluded, i.e. $E_{smoothness} = 0$, we assign a constant penalty λ_{occ} to any occluded segment. The second term $E_{occlusion}$ in Equation 4.2 accounts for the case of occlusion in the manner:

$$E_{occlusion} = \sum_{i=1}^{M} P_{occ}(s_i) \tag{4.4}$$

where

$$P_{occ}(s_i) = \begin{cases} \lambda_{occ} & \text{if } s_i \text{ is occluded} \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

4.3.3 Optimization through Greedy Algorithm

Finding the optimal labelling that minimizes the energy function 4.1 is a non-trivial problem. Given M color segments, each of which has N distinct depth levels, there are a total of M^N possible solutions to the labelling ρ. Such a large solution space is indicative of the complexity of this optimization problem. Moreover, the basic idea of spreading depth estimates from overlapping regions into neighboring non-overlapping regions implies a slow updating procedure. These factors suggest that the optimization of the energy function by global search would be a computationally expensive process.

Instead, the depth propagation procedure, which spreads the reliable depth estimates from R_o into R_{non}, indicates an *optimal substructure*[2] suitable for a *greedy*

[2] A problem exhibits optimal substructure if an optimal solution to the problem contains optimal solutions to its sub-problems.

algorithm implementation. As demonstrated in the last chapter, the locality of the smooth connection criterion indicates that a locally optimal solution, calculated by a greedy algorithm, produces a reasonable mosaic result.

Algorithm 1 The greedy algorithm pseudocode to calculate $\rho_{M1} = \{\rho_{M1}(s_i), i \in [1, M1]\}$, given inputs of $S_{M1} = \{s_1, s_2, \ldots, s_{M1}\}$

```
$\rho_{M1}^{new}(s_i) = \rho_{M1}^{old}(s_i) = occluded, i = [1, M1]$
$E(\rho_{M1}^{old}) = M1 \cdot \lambda_{occ}$
$mainloop = 1$
while ($mainloop < MaxLoop$) and ($\Delta E > Threshold$) do
  for $i = [1, M1]$ such that $s_i \in S_{M1}$ do
    for $j = [1, N]$ such that $d_j \in L$ do
      for $k = [1, M1]$ do
        $\rho_j(s_k) = \begin{cases} d_j & k = i \\ \rho_{M1}^{old}(s_k) & k \neq i \end{cases}$
      end for
      calculate $E(\rho_j)$
    end for
    $\rho_{M1}^{new}(s_i) = d_{\arg\min_j E(\rho_j)}$
  end for
  calculate $E(\rho_{M1}^{new})$
  if $E(\rho_{M1}^{new}) < E(\rho_{M1}^{old})$ then
    $E(\rho_{M1}^{old}) = E(\rho_{M1}^{new})$
    $\rho_{M1}^{old} = \rho_{M1}^{new}$
  end if
  $\Delta E = |E(\rho_{M1}^{old}) - E(\rho_{M1}^{new})|$
  $mainloop = mainloop + 1$
end while
$\rho_{M1} = \rho_{M1}^{old}$
```

The greedy algorithm starts with $S_{M1} = \{s_1, s_2, \ldots, s_{M1}\}$, the group of color segments immediately adjacent to the overlapping regions, and calculates the labelling $\rho_{M1} = \{\rho_{M1}(s_i), i \in [1, M1]\}$ that minimizes their energy as defined in Equation 4.1. The initial depth value for each $s_i \in S_{M1}$ is *occluded* and the initial energy of the entire group is $E(\rho_{M1}) = M1 \cdot \lambda_{occ}$. As illustrated in Algorithm 1, the calculation proceeds as follows, repeating until either the number of iterations exceeds a certain threshold or the change of labelling energy between iterations becomes insignificant.

1. For each color segment $s_i \in S_{M1}$, all its depth candidates, $d_j \in\ L$, are tested and the corresponding $E(\rho_j)$ are calculated with the depth estimates of all the other color segments fixed. The best depth value for s_i that minimizes the labelling energy is then chosen and recorded.
2. Once the best depth candidate has been found for each of the color segments in the group, if the total labelling energy, $E(\rho_{M1})$, is reduced with the new depth assignments, the depth estimates and the labelling ρ of the entire group are updated accordingly. Otherwise, these are left unchanged.

This process is applied in a similar manner to the neighbors of the previously processed group of segments, for which depth estimates have not yet been computed. This continues until no unprocessed color segments remain. Once all the depth estimates are obtained, the mosaic in R_{non} is rendered by copying the corresponding color segments from intermediate images into the mosaicing image plane to build the final panorama.

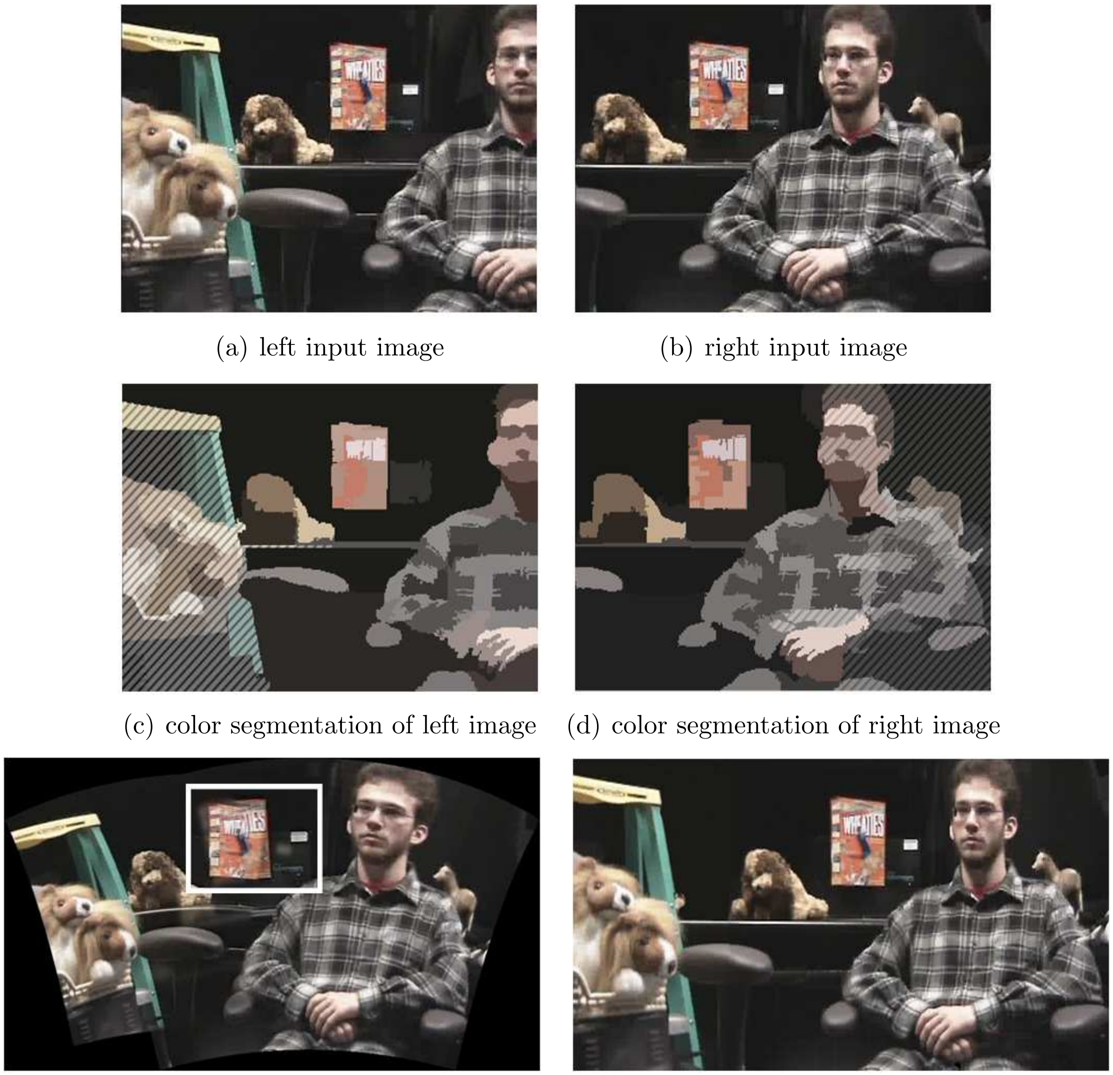
4.4 Experimental Results

The method is validated using the same data sets as in Section 3.3. The sample results provided here use inputs from two fixed video cameras with a large baseline, and thus exhibit non-trivial parallax effects. As was the case for our earlier approach, described in Section 3.3, we assume the images used in our experiments exhibited overlap over a minimum of 1/3 of their areas. These results can be extended in a straightforward manner to take advantage of additional cameras. The inputs need not necessarily be rectified; a translationally dominant camera configuration accompanied by some panning or tilting rotations is also acceptable.

As seen in Figure 4–1e, the complicated depth distribution in the data set from Seitz [49] results in obvious distortion (e.g., of the human subject) and artifacts (e.g., the ghost errors of the box and the light switch) in the Autostitch mosaic result. In contrast, the depth-based image mosaicing (DBM) technique described in this chapter synthesizes a reasonable panorama, as shown in Figure 4–1f. The noticeable improvements of the DBM technique over traditional image mosaicing algorithms is illustrated once more in Figure 4–2 using our own data set.

When presented with sparse inputs containing non-trivial parallax effects, both the DBM method and the virtual dense sampling image mosaicing algorithm (VDS), introduced in Chapter 3, generate reasonable mosaic results. These satisfy the important perceptual properties described in Section 1.1. We note, however, the fundamental difference that the VDS generates multi-perspective mosaic results while the DBM renders the panorama from the perspective of a single virtual reference camera. For both the VDS and DBM, the computationally most expensive processing step is the synthesis of virtual frames. Because the DBM needs to calculate only one virtual frame, whereas the VDS needs to perform this calculation once for each of the virtual samples[3] that are synthesized, we conclude that another significant difference is that the computational expense of the DBM is much lower than that of the VDS.

[3] In our case, we use 50 virtual samples, as discussed in Section 3.1.3



(a) left input image

(b) right input image

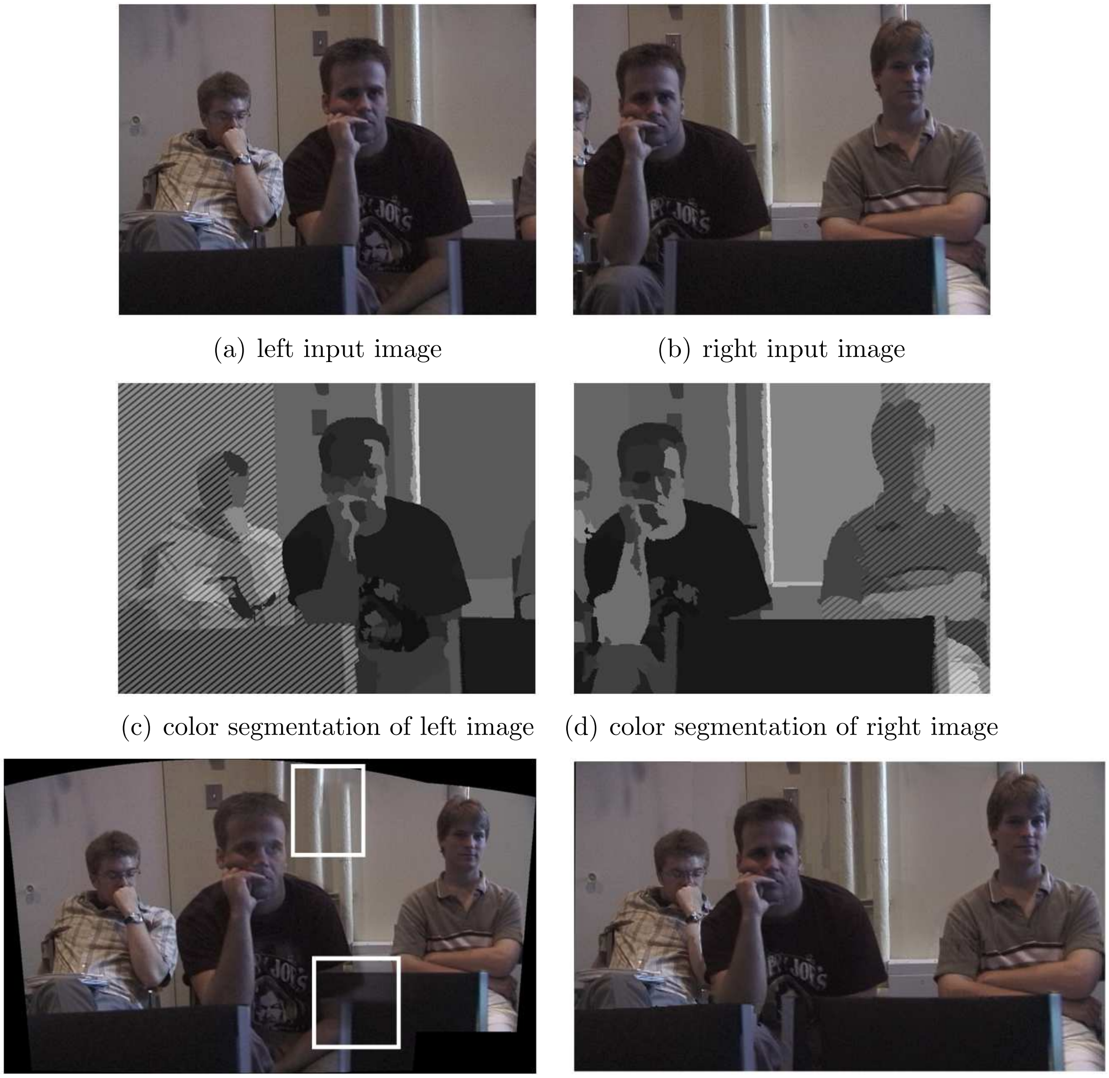
(c) color segmentation of left image

(d) color segmentation of right image

(e) mosaicing result of Autostitch

(f) depth-based image mosaicing result

Figure 4–1: Depth-based image mosaic results of data set from Seitz. The virtual mosaicing camera is chosen to coincide with rightmost source camera. (c) and (d) Hashed regions of the color segmentations contribute to non-overlapping regions in the final mosaic. (e) Autostitch mosaic result exhibits distortion and artifacts as indicted by the overlaid rectangle. (f) Our algorithm synthesizes a much improved result.



(a) left input image

(b) right input image

(c) color segmentation of left image

(d) color segmentation of right image

(e) mosaicing result of Autostitch

(f) depth-based image mosaicing result

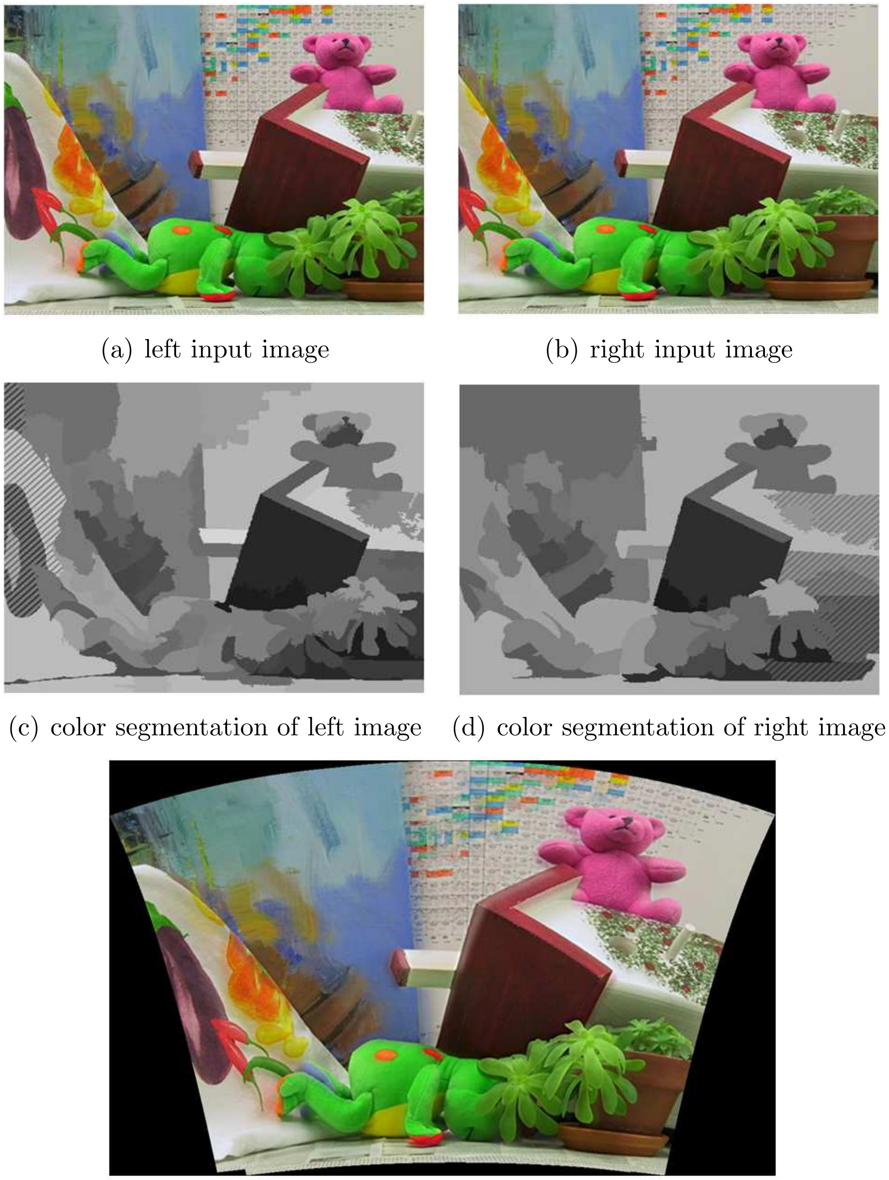
Figure 4–2: Depth-based image mosaic results of our own data set. The virtual mosaicing camera is chosen to coincide with rightmost source camera. (c) and (d) Hashed regions of the color segmentations contribute to non-overlapping regions in the final mosaic. (e) Autostitch mosaic result exhibits distortion and artifacts as indicted by the overlaid rectangles. (f) Our algorithm synthesizes much improved result.

For further comparison, a reference mosaic generated with the DBM method using the ground truth depth values from the teddy data set is shown in Figure 4–4b. The virtual mosaicing camera is chosen to coincide with the left source, i.e., the camera that captures the second image (*im2*) in the data set, and its internal parameters are adjusted to accommodate more image contents within a wider FOV. The overlapping regions in our mosaic result, i.e., the regions within the black boundaries in Figure 4–4a, exhibit reasonable coherence with the reference mosaic. However, slight appearance differences due to the variance of depth estimates are observed in the non-overlapping regions, i.e., the regions outside the black boundaries, especially toward the right of the mosaic result in Figure 4–4a.

It bears comment that unlike image-based rendering algorithms [51], our DBM approach does not attempt to determine the real depth in non-overlapping regions. Indeed, with the naive assumption of uniform depth of each color segment in R_{non}, the DBM method generates depth estimates that usually do not conform to the ground truth topology of most scenes. Nevertheless, the depth estimates obtained though a smooth appearance connection criterion guarantee sufficient resemblance between local regions in the mosaic results and those in the inputs. Thus, outputs based on these depth estimates still appear reasonable and perceptually acceptable.

4.5 Limitations

The quality of mosaic results is in large part dependent on the quality of color segmentation. As explained in Section 4.3.1, provided this segmentation preserves



(a) left input image

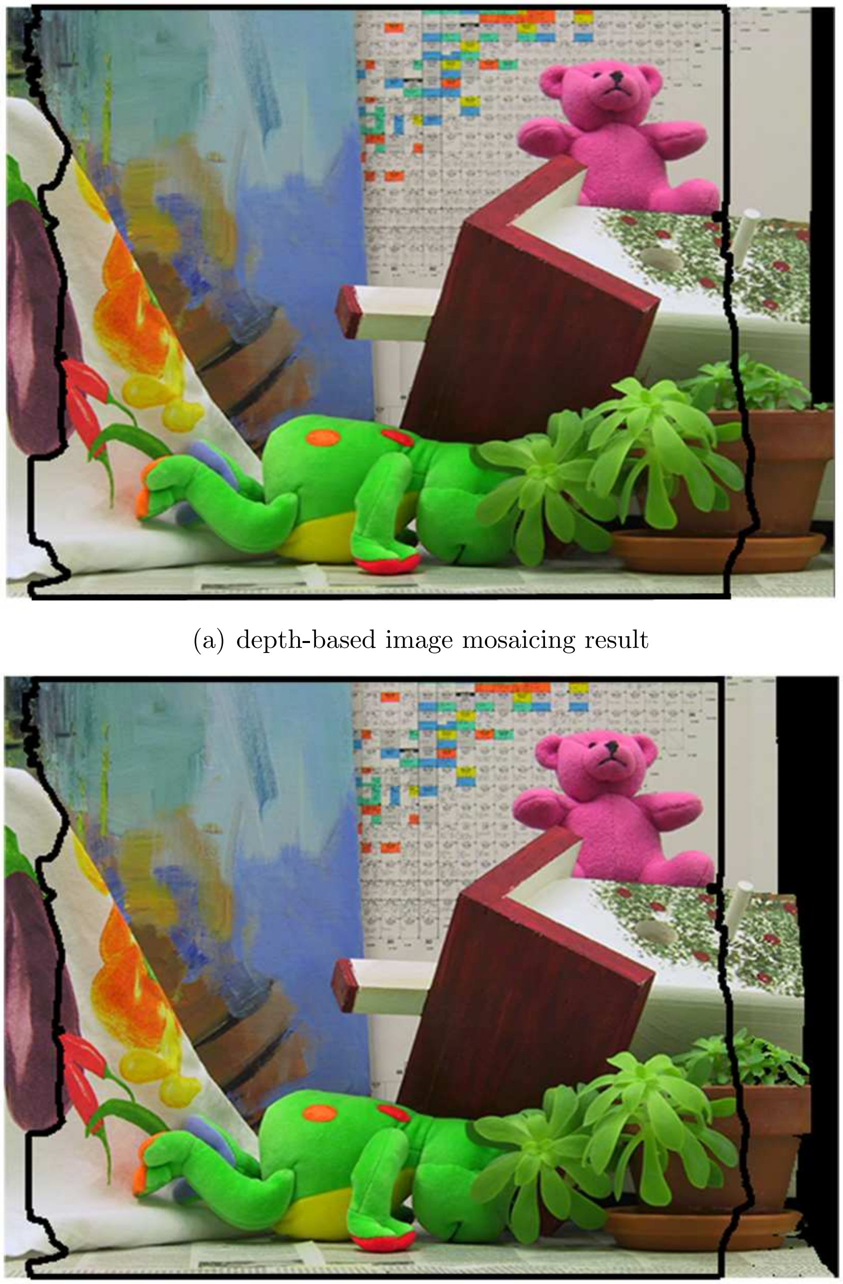
(b) right input image

(c) color segmentation of left image

(d) color segmentation of right image

(e) mosaicing result of Autostitch

Figure 4–3: Comparison with the reference mosaic using the ground truth depth values of Teddy data set (part I). The virtual mosaicing camera is chosen to coincide with the left source camera. (c) and (d) Hashed regions of the color segmentations contribute to non-overlapping regions in the final mosaic. (e) Autostitch mosaic result exhibits distortion and artifacts.



(a) depth-based image mosaicing result

(b) DBM result using ground truth depth value

Figure 4–4: Comparison of (a) our DBM result and (b) reference DBM result constituted using the ground truth depth values of Teddy data set. In both results, the areas within or outside the black boundaries correspond to the overlapping and non-overlapping regions, respectively.

edges well along depth discontinuities, larger color segments are preferred to oversegmentations. A careful tuning procedure of the color segmentation algorithm proposed by Georgescu *et al.* [21] is required for a specific separation of color segments in order to obtain the best output mosaic for a given data set. The need for this empirical tuning limits the system's suitability to many real-life applications. Furthermore, ever with a carefully tuned color segmentation, occlusions in R_{non} may still induce undesirable artifacts.

If the virtual mosaicing camera has different position and orientation from those of the source camera, which covers the non-overlapping regions, holes due to occlusion may occur in R_{non} of the mosaic result. These holes do not have any correspondence in the inputs, and thus violate the smooth appearance connection criterion, i.e., a local region in the output mosaic should resemble some corresponding region in the sources. Filling these hole regions using the DBM method based on the smooth appearance connection criterion may prove insufficient unless the holes actually correspond to textureless scene content.

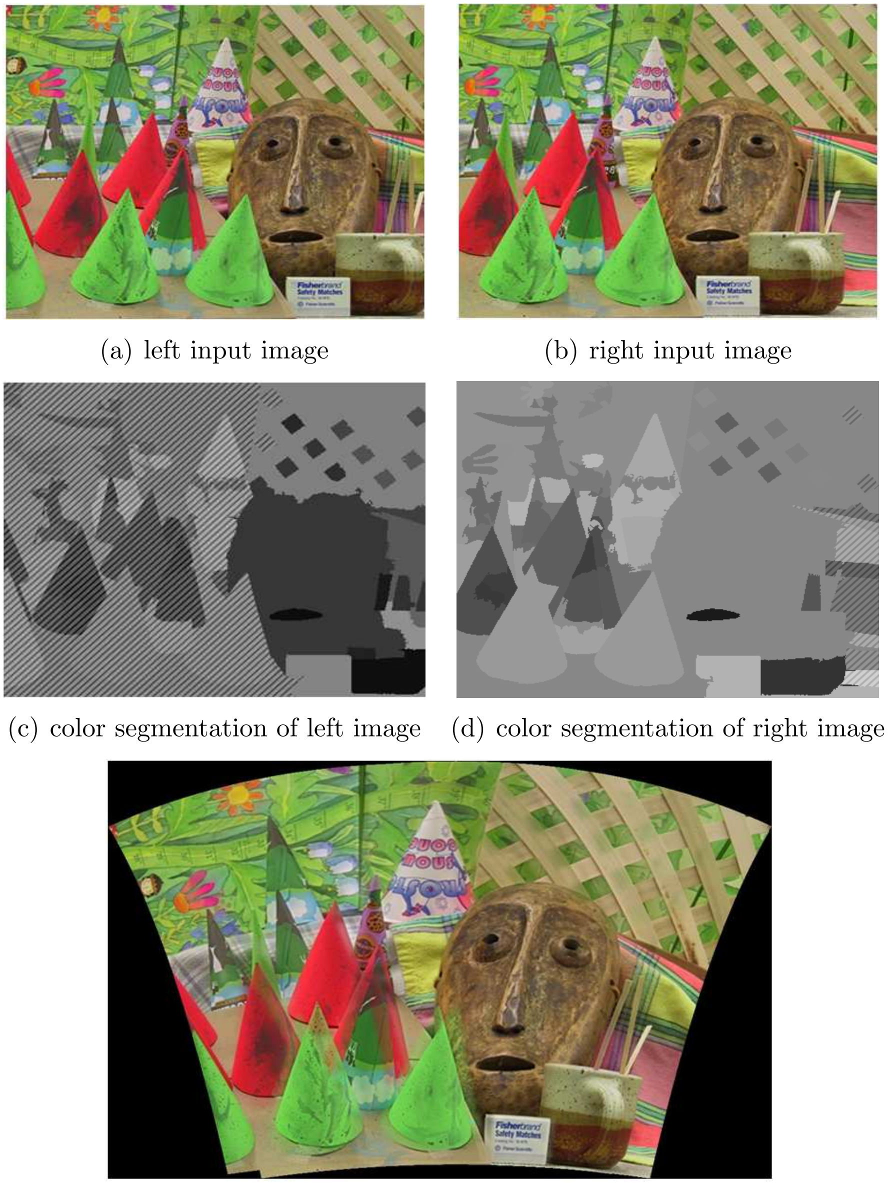
We use the *Cone* data set [48] as a more challenging input set to test the limitations of our algorithm with regard to occlusions. Furthermore, we deliberately restrict the algorithm to employ only half of the mosaic in the overlapping regions calculated by the plane sweep algorithm, from which depth estimates are propagated to the other half, exhibiting more occlusions. As expected, noticeable holes appear in the mosaic result (Figure 4–6a). Because of the texture in the corresponding scene contents, filling these holes may result in artifacts. However, it should be noted that holes also appear in the reference mosaic result using the ground truth depth values (Figure 4–6b).

Notwithstanding the holes due to occlusion, the DBM method constructs much improved results compared to traditional algorithms, as shown in Figure 4–1, 4–2, 4–3 and 4–5, with respect to its avoidance of distortions and synthesis artifacts resulting from the image alignment errors. Although we have not characterized quantitatively the maximum texture complexity and depth complexity that the DBM method can tolerate,[4] for all our experiments in this thesis, the input images in this *Cone* data set exhibit the most complicated texture and depth distributions, which are seldom encountered in the real world cases. In general, the new DBM technique introduced in this chapter overcomes the parallax problem and produces reasonably high quality panoramas for most real-world applications.

4.6 Summary

This chapter describes our second solution to overcoming parallax issues in image mosaicing. This approach explicitly uses depth cues to render scene contents, whether from overlapping or non-overlapping regions of the source data, onto their proper positions in the final mosaic. Experimental results indicate the strengths of the new algorithm in generating mosaics that are generally free of parallax-related artifacts, even when presented with challenging inputs.

[4] This is a long-lasting unsolved problem common to all the new view synthesis algorithms.



(a) left input image

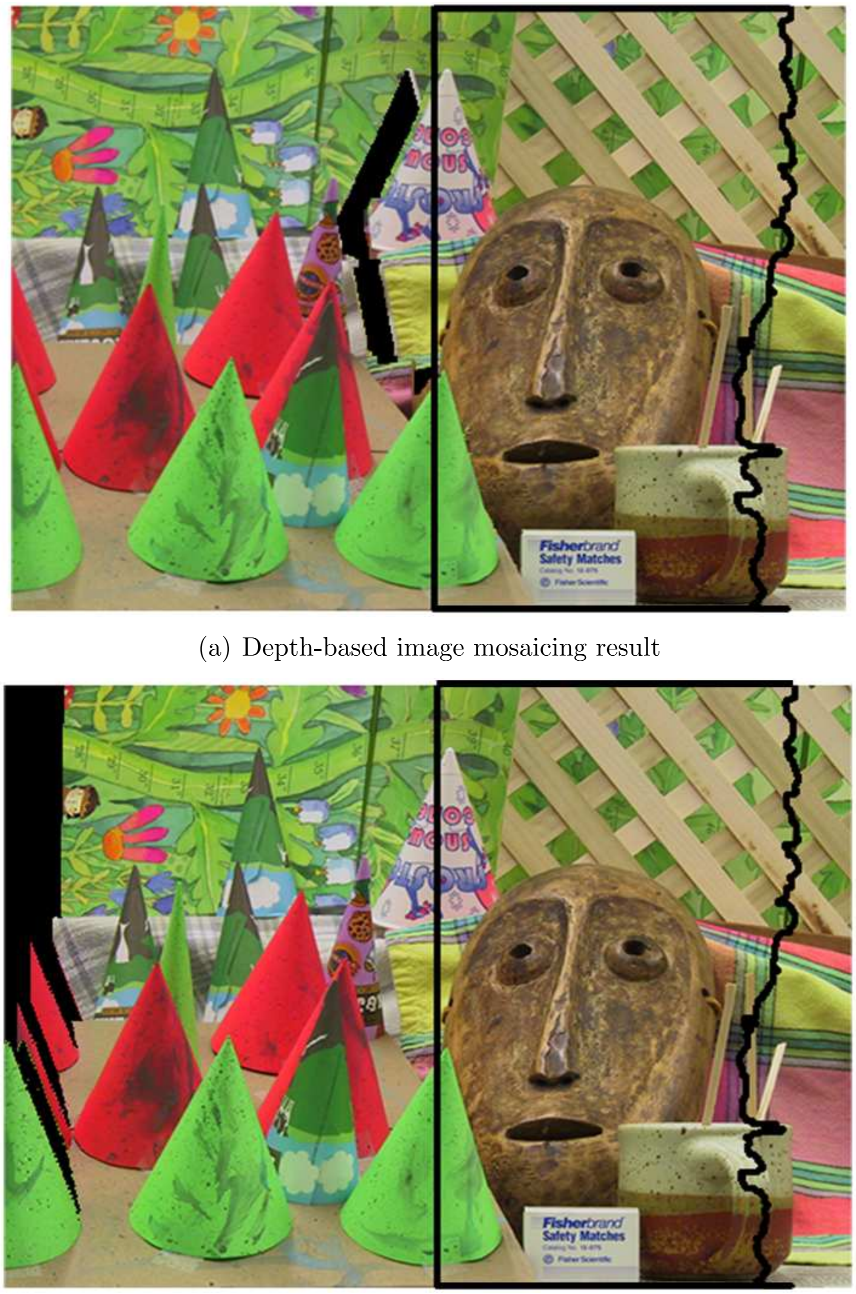
(b) right input image

(c) color segmentation of left image

(d) color segmentation of right image

(e) mosaicing result of Autostitch

Figure 4–5: Comparison with the reference mosaic using the ground truth depth values of Cone data set (part I). The virtual mosaicing camera is chosen to coincide with the right source camera. (c) and (d) Hashed regions of the color segmentations contribute to non-overlapping regions in the final mosaic. (e) Autostitch mosaic result exhibits distortion and artifacts.



(a) Depth-based image mosaicing result

(b) DBM result using ground truth depth value

Figure 4–6: Comparison of (a) our DBM result and (b) reference DBM result constituted using the ground truth depth values of Cone data set. In both results, the areas within or outside the black boundaries correspond respectively to the overlapping and non-overlapping regions.

As an improvement over the the virtual dense sampling approach introduced in the previous chapter, this algorithm requires significantly reduced computation for mosaic synthesis of static scenes. Furthermore, it offers the potential to address mosaicing problems involving dynamic scenes, as will be discussed in details in the following chapter.

CHAPTER 5
Depth Based Dynamic Mosaicing

5.1 Introduction

We now elaborate the solution to the problem of generating a perceptually correct mosaic that includes moving objects in the scene.[1]

To ensure continuous acquisition of video in both time and space over the desired viewing area, a single moving video camera, as used in the majority of previous dynamic mosaicing research [4][6][24], is clearly insufficient. Instead, we use a multiple camera configuration with a large baseline, which poses a greater challenge to any mosaicing system because of its non-parallax-free character.

As discussed in Section 4.1, obtaining accurate image alignment, or registration, usually the first step in traditional image mosaicing techniques, remains difficult when depth variance in the scene causes obvious parallax effects in the inputs. Furthermore, it proves difficult to preserve the temporal coherence of these image registration results when objects at different depths to the camera are dynamically changing their gestures and positions.

These issues of misalignments lead to several potential problems for traditional image mosaicing techniques to generate dynamic mosaics when presented with inputs

[1] The work in this chapter is based on the contents of the author's Oceans and ICPR publications [41][43].

containing moving objects. First, the static background may appear to distort, shift, or jitter over time. Second, the motion of foreground objects may be inconsistent, as if resulting from a sudden change of camera parameters. Third, the parallax effects in inputs may cause noticeable ghost errors about either the static or moving contents in dynamic mosaicing results.

To cope with these issues, our depth-based image mosaicing (DBM) approach performs a segmentation of foreground and background layers using a Mixture-of-Gaussians (MoGs) model, and then projects these layers separately onto the mosaicing plane, according to their respective depth estimates, to render the final result. This guarantees both temporal and spatial coherence of the resulting mosaic video.

The remainder of this chapter is organized as follows: The MoGs modelling based segmentation algorithm is discussed in Section 5.2, followed by an introduction of background mosaic construction in Section 5.3. The foreground mosaic construction in overlapping regions and non-overlapping regions are discussed in Sections 5.4 and 5.5 respectively. Section 5.6 provides a comparison of experimental results with those of Autostitch. Finally, Section 5.7 summarizes this work and its implementations.

5.2 Foreground-background Segmentation

Foreground-background segmentation is a process of separating objects of interest, such as the moving human subjects in our cases, from the rest of the image, i.e., the static background. The process can be categorized as either motion-based, depth-based or stochastic background modelling based, according to the method of image measurement.

Motion-based segmentation algorithms analyze either optical flow [10][53] or differences between consecutive image frames [30][61] to determine which pixels exhibit motion and thus, are classified as moving foreground objects. However, the impact of shadows and illumination changes are usually not considered by motion-based algorithms. Depth-based segmentation algorithms [26][27][35] detect foreground objects, for example, a human subject standing in the scene, as pixels whose depth values differ from the expected background. These algorithms are robust to photometric variance of the background; however, their dependence on stereo information prevents them from processing input corresponding to non-overlapping regions of the scene.

Inspired by the observations that foreground human objects are rarely stationary and usually have distinct appearances from the background, a stochastic background model may provide a means of distinguishing between the rapidly changing foreground contents and the slowly varying background scene. The work of Stauffer and Grimson [54], who modeled each pixel by a mixture of Gaussians (MoGs), which were updated adaptively, online, according to slow changes of background, is a standard background subtraction technique. In our work, we apply an improved version by Lee [32], which used an adaptive, rather than fixed, learning rate for each Gaussian to obtain improved convergence speed without sacrificing stability. This process is explained in further detail in the following sections.

5.2.1 Mixture-of-Gaussian Background Model Construction

Suppose all the pixels from the frames in a given time interval satisfy the distribution of a Mixture-of-Gaussians (MoGs) background model. In this case, the probability that a pixel assumes a value X is given by Stauffer and Grimson [54]:

$$P(X) = \sum_{i=1}^{K} \omega_i P(X|G_i) \tag{5.1}$$

where

$$P(X|G_i) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_i|}} \exp -\frac{1}{2}(X - \mu_i)^T \Sigma_i^{-1} (X - \mu_i) \tag{5.2}$$

where μ_i is the mean of the i'th Gaussian, Σ_i is the diagonal covariance matrix, and ω_i is its weight.

Any new pixel value is compared against the available models. If the value can be represented by an element of the MoGs, it is used to update the model. Otherwise, the least-likely Gaussian element, i.e., with the smallest ω_i, is replaced by a new Gaussian initialized with the new pixel value. Our present implementation uses $K = 3$ Gaussians.

For our purposes, pixel values are represented in the YC_bC_r colour space, because this offers greater robustness to photometric noise than either RGB or HSI [31]. Source frames, which are used to train the MoGs model, are not required to contain only pure background scenes. In fact, samples including moving foreground help the model learn the characteristics of dynamic shadows, which should be classified as part of the background.

5.2.2 Segmentation

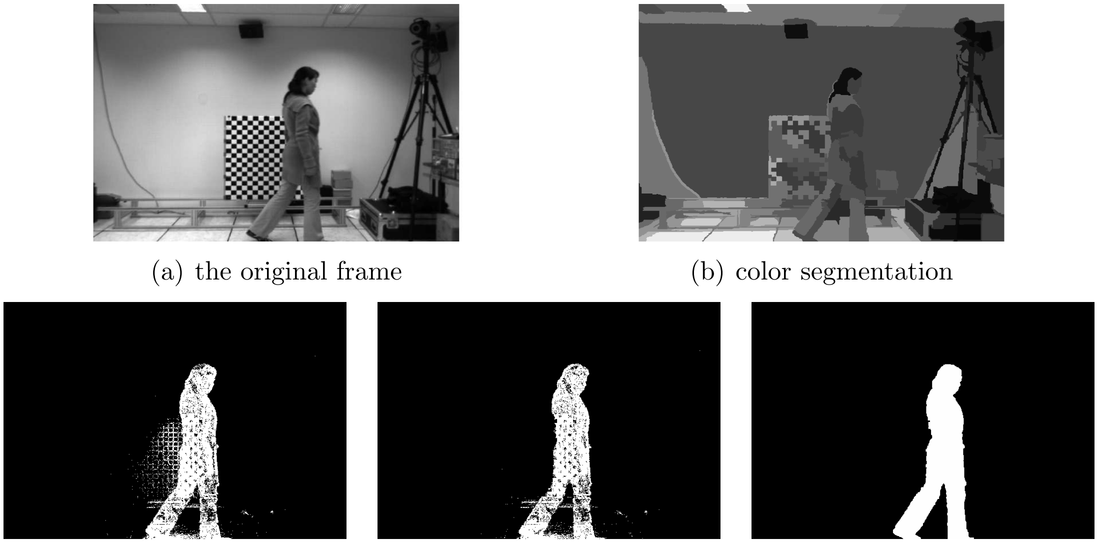
For a given frame, the probability that each pixel, X, belongs to the background, B, is calculated according to the trained MoGs background model:

$$\begin{aligned} P(B|X) &= \sum_{i=1}^{K} P(B|G_i)P(G_i|X) \\ &= \frac{\sum_{i=1}^{K} P(X|G_i)P(G_i)P(B|G_i)}{\sum_{j=1}^{K} P(X|G_j)P(G_j)} \end{aligned} \tag{5.3}$$

where $P(B|G_i) = 1/(1 + e^{-aw_i/\sigma_i} + b)$ for $G_i, i = [1, 3]$, as defined by Lee [32]. a and b are constants, ω_i is the weight of i'th Gaussian, and σ_i is the average of the summation of the trace in Σ_i. If the probability $P(B|X)$ exceeds some threshold, the pixel is considered as an element of the background, and otherwise, as a foreground (or non-static) object. Thus the output of the segmentation process is a binary mask with white pixels representing foreground regions and the remaining black area corresponding to the static background.

Dynamic shadows cast by moving objects onto the background surfaces may pose a challenge to the segmentation algorithm as these may lead to false-positive detection as foreground regions. In our experimental environment, because the short distance between a moving object and the static checkboard results in a significant reduction of light reaching the surface, and thus, shadows on the checkboard are obvious, as in Figure 5–1c.

A post-processing step of shadow removal, which takes advantage of the simple color of the checkboard, is applied to correct the segmentation errors. First, we calculate the mean (μ) and variance (σ) of the color distribution for every pixel on



(a) the original frame

(b) color segmentation

(c) raw mask

(d) after shadow removal

(e) final cleaned mask

Figure 5–1: An illustration of foreground and background segmentation. (c) The raw mask of the foreground layer contains some dynamic shadow points cast on the checkboard. (d) These shadow points are removed. (e) The final mask after integrating color segmentation information from (b).

the checkboard. Then, in a given frame, if the color of any assumed checkboard pixel is within a $3\sigma^2$ distance from μ, it is removed from the segmentation result. In this way, shadow pixels in the initial segmentation results, after thresholding $P(B|X)$, are largely eliminated, as illustrated in Figure 5–1d.

The raw mask after thresholding and shadow removal contains many isolated foreground points. Combining the result of color segmentation of the input frame, as shown in Figure 5–1b, we integrate these isolated points into a group of color segments contributing to a silhouette of a foreground object with boundaries closely matching the outlines of a human body, as illustrated in Figure 5–1e. This silhouette will later assist in the generation of foreground mosaicing frames.

5.2.3 Background Image Generation

In addition to the binary foreground-background segmentation result, the *background image*, a weighted sum of the means of each Gaussian element from the MoGs, is also constructed. This provides a full description of the static regions in the environment whereas the complement of a silhouette is incomplete due to occlusions from foreground objects.

[2] If a data distribution is approximately normal, then approximately 68% of the values are within 1σ from the μ, approximately 95% of the values are within 2σ and approximately 99.7% lie within 3σ. This is known as the 68-95-99.7 rule, or the empirical rule. In our case, we chose 3σ to distinguish checkboard pixels from the others.

The background image is computed as the expected value, $E[X|B]$, of the observation X, assuming it to be background, as the following weighted average [32]:

$$\begin{aligned} E[X|B] &= \sum_{i=1}^{K} E[X|G_i]P(G_i|B) \\ &= \frac{\sum_{i=1}^{K} \mu_i P(B|G_i)P(G_i)}{\sum_{j=1}^{K} P(B|G_j)P(G_j)} \end{aligned} \tag{5.4}$$

where μ_i represents the mean of the i'th Gaussian, G_i.

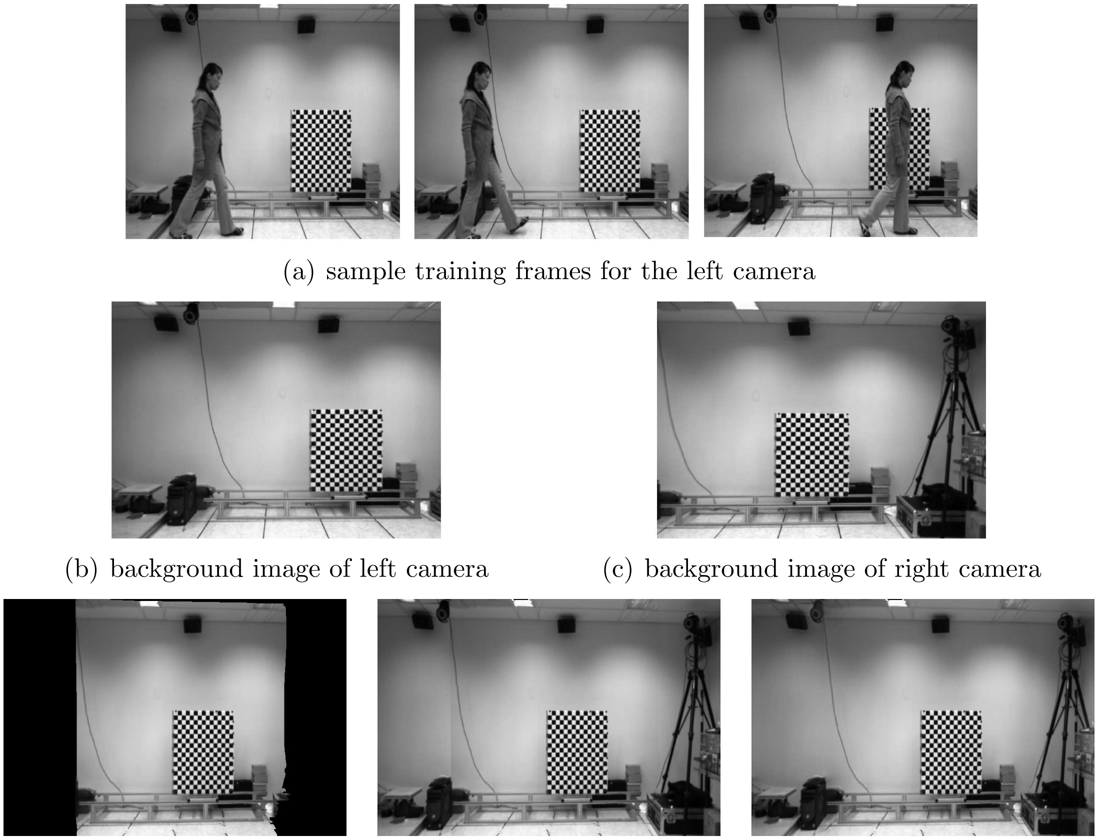
Two such background images are pictured in Figure 5–2b and 5–2c. These are used to construct the background mosaic as explained in the next section.

5.3 Mosaic Construction of Background Layer

As discussed at the beginning of this chapter, in order to preserve both the spatial and temporal coherence of dynamic mosaicing video, the static background and the moving foreground should be processed separately. We first describe the procedure to construct a background mosaic that is common to all frames.

The pose and internal parameters of the virtual camera are chosen according to the method shown in Section 4.2. Using the depth-based image mosaicing (DBM) technique introduced in the last chapter, both overlapping and non-overlapping regions of the mosaic are rendered as if seen by this virtual camera, which has all its parameters fixed during the synthesis of the entire dynamic mosaicing video.

Figure 5–2 illustrates the background images and the depth-based mosaic based on them. Because the dynamic foreground objects have been removed, they exercise no influence over the image alignment process, and thus, the appearance of the background in the mosaic video remains consistent over time. In our tests, a multi-band



(a) sample training frames for the left camera

(b) background image of left camera

(c) background image of right camera

(d) depth-based background mosaic of overlapping region

(e) after non-overlapping region filled

(f) after multi-band blending

Figure 5–2: Mosaic construction of background layer.

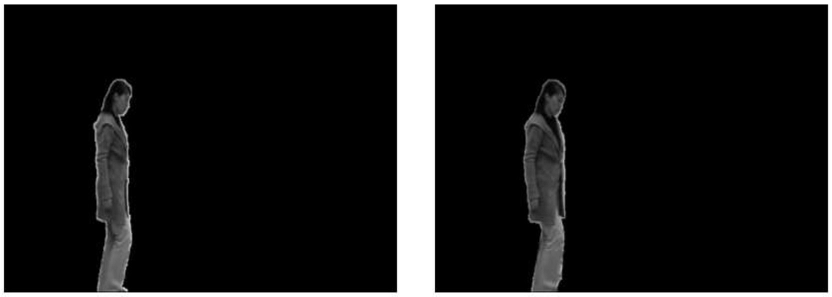
blending strategy [14] is required to balance the color differences between camera responses. This smooths the boundary between overlapping and non-overlapping regions in the background mosaic.

5.4 Foreground Mosaic in Overlapping Regions

Similar to the work in Section 4.2, generating a depth-based image mosaic of the foreground layer in the overlapping regions, R_o, of multiple input cameras, is a typical view synthesis problem. Given the set of input frames and their corresponding camera parameters, the plane sweep algorithm [16] and graph cut optimizations [8][9] are applied to construct a synthesized foreground mosaic in R_o observed from the virtual mosaicing camera.

When foreground objects move beyond the FOV of a single input camera, we have to combine the foreground layer contents from multiple sources to build a complete mosaic video. Under this condition, the large baseline separation between cameras, either real or virtual, may result in an appearance problem of the foreground layer. As illustrated in Figure 5–3, an object leaves R_{non} and enters R_o between two successive frames. When switching between the foreground layer from source camera C_1, in R_{non} to that of a virtual mosaicing camera, C_v, it is difficult to achieve a smooth transition of foreground content due to the different viewpoints of the two cameras. This results in an obvious difference of object pose as seen in Figure 5–3.

Our solution is to synthesize foreground layers at each time instance within $[t+1,\ t+W]$, from W smoothly interpolated positions between cameras C_1 and C_v, where $W = 10$ in our experiments. We then project each of these synthesized foreground layer instances onto the mosaicing plane according to their depth estimates so as to



(a) foreground mosaic at time t, when object is in R_{non} (b) foreground mosaic at time $t+1$, when object enters R_o

Figure 5–3: The obvious appearance differences between foreground layers when an object leaves R_{non} at time t and enters R_o at time $t+1$.

build dynamic mosaic frames containing only foreground object. In such a manner, the abrupt changes of foreground appearance over two successive frames at time $[t,\ t+1]$, as illustrated in Figure 5–3, are replaced by gradual and smooth transitions over $[t,\ t+W]$ frames, as presented in Figure 5–4.

5.5 Foreground Mosaic in Non-overlapping Regions

Since no stereo information is available to estimate depth values of foreground objects in the non-overlapping regions, R_{non}, depth-based foreground mosaics in these regions must be processed differently from that in R_o.

5.5.1 Previous Approaches of Depth Calculation

View synthesis techniques based on stereo information are obviously not suitable for calculating depth values of foreground layer contents in R_{non}.

However, when presented with a monocular video sequence, the technique of structure-from-motion may reconstruct the shape and position of moving objects. The method stems from the factorization algorithm by Tomasi and Kanade [58],

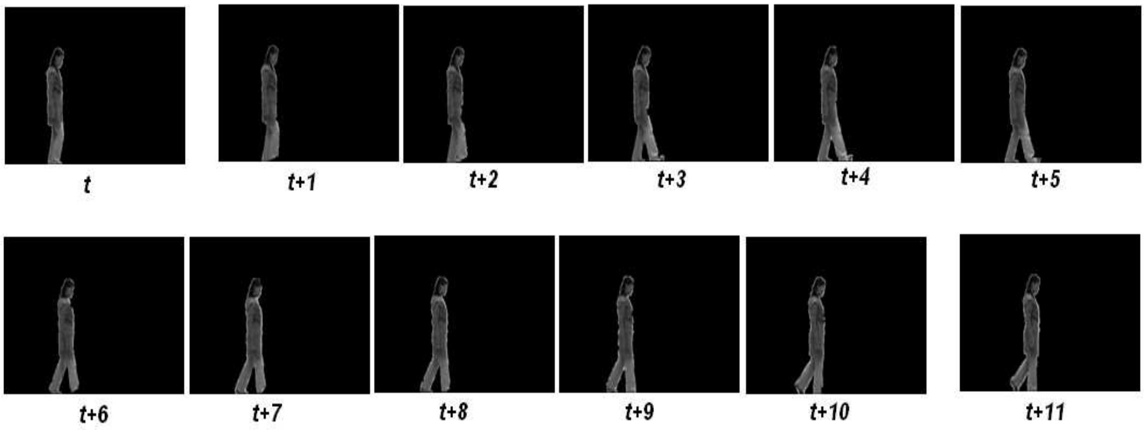


Figure 5–4: The illustration of the usage of interpolated virtual mosaic frames to compensate for an abrupt appearance difference between foreground layers, when the person moves from R_{non} to R_o, as presented in Figure 5–3. We synthesize 10 foreground layers from time $t+1$ to $t+10$ respectively. We also present the foreground mosaic frames at time t and $t + 11$ to illustrate the smooth transition between this group of interpolated virtual mosaic frames with the rest of the dynamic mosaicing video.

which was only applied to rigid objects. The basic idea is to decompose a measurement matrix, which contains image coordinates of a group of matching features across the entire video sequence, into its components of shape and motion. The former represents the $3D$ positions of the feature points included in the measurement matrix, and the latter describes the relative movement between a video camera and scene objects.

A newer technique, nonrigid structure-from-motion [11][18][59] can estimate time-varying $3D$ shapes from $2D$ point tracks in monocular video input but is still limited to objects with comparatively low degrees of freedom of movement. However, a full-body human subject may generate significantly more complex motion patterns, and produce self-occlusions during movement. These factors make $3D$ reconstruction of such an object from monocular video difficult. Reconstruction may even become ill-posed if arbitrary deformations are allowed.

Instead of considering other approaches to discover the real $3D$ shape and position of foreground moving objects in R_{non}, we focus on a perceptually correct representation of foreground motion in a dynamic video while preserving spatial and temporal motion consistency.

5.5.2 Motion Perception and its Mathematical Representation

Psychological research of motion perception reveals that humans are sensitive to significant changes of physical features, i.e., *speed* and *direction* in motion trajectories [28]. People understand motion activities based on when and where these changes occur [64]; moreover, the order of these events is another important cue for perception [38].

Rao *et al.* [45] translated these conclusions into a mathematical representation. A $2D$ trajectory, which represents the path of an object in a video sequence, is defined as a spatiotemporal curve with the function:

$$r(t) = [x(t),\ y(t),\ t] \quad 1 \leq t \leq n \tag{5.5}$$

where t represents the frame index and $[x(t), y(t)]$ indicates the pixel coordinates of the object centroid in the t'th frame. The curvature, $k(t)$, which is responsive to discontinuities in velocities and accelerations of $r(t)$, is given by:

$$k(t) = \frac{\|r'(t) \times r''(t)\|}{\|r'(t)\|^3} \tag{5.6}$$

where $r'(t)$ or $r''(t)$, the first and second derivatives of $r(t)$, represent its velocity and acceleration respectively.

Significant changes of physical motion features, also named *instants*, are defined as the maxima in curvature of a $2D$ trajectory [45]. Each $2D$ trajectory can be decomposed into a sequence of instants, which is the mathematical representation of motion understood and distinguished by humans.

As long as cameras remain in the same half hemisphere of the viewing space, such instants are independent of view direction. In other words, although the $2D$ trajectories of the same motion may appear differently from various camera positions, as illustrated in Figure 5–5, they share a common sequence of instants, which leads to the identical perception of object movement.

In the next section, we discuss the details of our novel algorithm. This generates perceptually correct dynamic mosaicing video by propagating the reliable depth

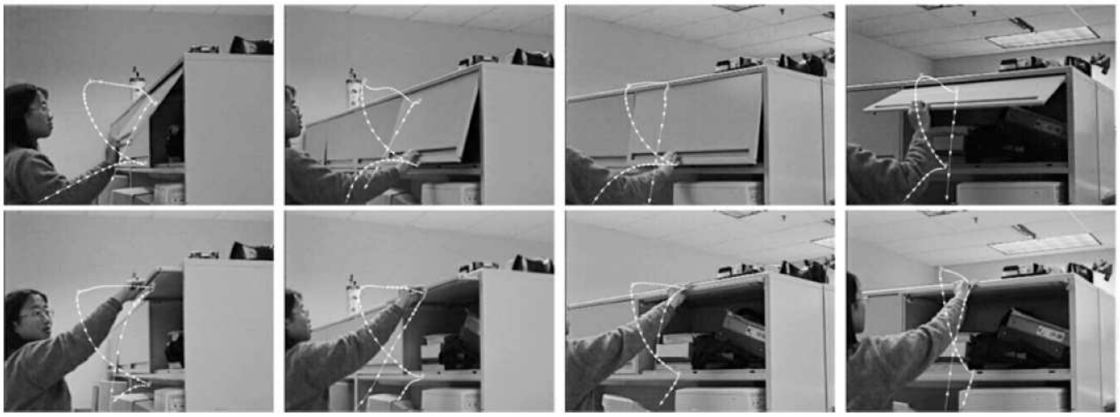


Figure 5–5: Trajectories, the paths consisting connected dots in white, from different view points for opening (top) and closing (bottom) overhead cabinet action. Both the opening and closing actions in the same column are taken at the same viewpoint. Figure 10 of Rao *et al.* [45] (©[2002]Springer, reproduced here with permission).

information of the foreground layer from overlapping regions into neighboring non-overlapping regions, in such a manner that the sequence of motion instants is preserved.

5.5.3 Problem Definition

Our goal is to synthesize a mosaic video by generating a sequence of foreground mosaic frames in the non-overlapping regions, from the perspective of the virtual mosaicing camera.

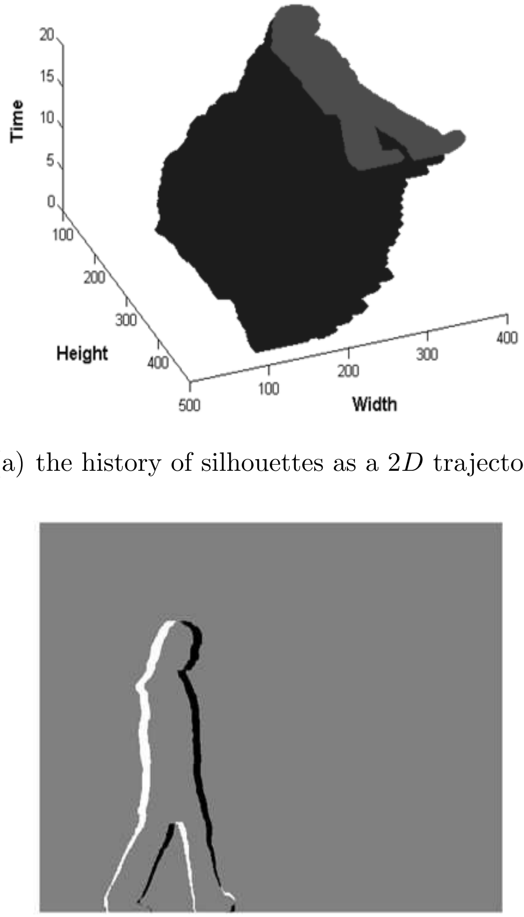
As discussed above, humans perceive motion through instants, i.e., the locations of curvature maxima in a $2D$ trajectory, $r(t)$. If the motion in an output mosaic video is generated by naively replicating the velocity of $r(t)$ in the monocular input video, curvature is preserved, as this depends on the velocity and its first derivative. As such, the sequence of instances, i.e., the perception of motion, is also preserved.

Furthermore, motion consistency, as observed in the input video, is achieved in both spatial and temporal domains in the output mosaic video.

It bears comment that, the output constructed in this manner does not faithfully present the projection of real $3D$ motion seen by the virtual mosaicing camera. However, given the fact that it is computationally expensive, or sometimes, even impossible to achieve reliable $3D$ information when presented with a monocular input video, we consider it acceptable to obtain output that maintains the correct perception of motion and continuity of motion. As in the case of static mosaicing, for which we require a smooth transition of appearance between different sources, here, for dynamic mosaicing of foreground layers, we require consistent perception of motion as objects move across the FOVs between different cameras.

For simplicity, we ignore the depth variance of foreground objects, treating them as if parallel objects to the image plane of the viewing camera. Consequently, constructing a foreground mosaic video in R_{non} becomes a procedure to find a suitable depth level d for each foreground object at every time t. After projecting these onto the mosaicing image plane according to their depth estimates, the motion trajectory of these warped foreground objects, $r_{new}(t)$, should present the same velocities as those of $r(t)$ in the monocular input video.

Let L denote the set of depth levels $\{d_1, \ldots, d_N\}$. In our application, the $2D$ motion trajectory in the input video, $r(t)$, is represented by the history of silhouettes, as shown in Figure 5–6a. The $2D$ motion trajectory in the synthesized mosaic video, $r_{new}(t)$, consists of the history of warped silhouettes. The velocity $v(t)$ of a $2D$ trajectory is related to the first derivative of the silhouette sequence, i.e., $\Delta S(t) =$



(a) the history of silhouettes as a $2D$ trajectory

(b) the difference between two successive silhouettes

Figure 5–6: (a) An example of a $2D$ motion trajectory, which is represented by a history of silhouettes, containing $\{S(1), ..., S(20)\}$. (b) The velocity of a $2D$ motion trajectory is related to the difference between two successive frames, where regions in white are evaluated with value “-1”, regions in black with value “1”, and regions in gray with value “0”.

$S(t+1) - S(t)$ as shown in Figure 5–6b. The speed component of the velocity of $r(t)$ is given by:

$$Speed(\Delta S(t)) = \frac{\sum_{s(i,j)=1} \Delta S(t)}{\sum_{s(i,j)=1} S(t+1)} + \frac{\sum_{s(i,j)=-1} \Delta S(t)}{\sum_{s(i,j)=1} S(t)} \tag{5.7}$$

$Speed(\Delta S(t))$ is a sum of two area ratios, the first of which is between the area of regions with value "1" in $\Delta S(t)$ and that of the foreground layer in silhouette $S(t+1)$, while the second is between the area of regions with value "-1" in $\Delta S(t)$ and that of the foreground layer in silhouette $S(t)$. The direction of velocity is defined as:

$$\vec{\gamma}(t) = \text{sgn}(CM_1(t) - CM_{-1}(t)) \tag{5.8}$$

where CM_1 is the centroid of "1" valued regions and CM_{-1} is that of "-1" valued regions in $\Delta S(t)$. The function sgn(x) is given by:

$$\text{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 1 & \text{otherwises} \end{cases} \tag{5.9}$$

If $\mathbf{x}$ is a vector, sgn$(\mathbf{x})$ returns a vector constituted of the sgn values of each of its elements.

From these definitions, the problem of generating the t'th depth-based foreground mosaic frame in R_{non} can be stated precisely as follows:

Problem. *Given a monocular input video of a foreground object in R_{non} and assuming that the object is of uniform depth, find the best depth estimate $d(t)$, so that the difference of velocity, $dif_v(t)$, between $r(t)$ and $r_{new}(t)$ is minimized.*

The difference of velocity, $dif_v(t)$, is defined as follows:

$$dif_v(t) = dif_{speed}(t) + dif_{direction}(t) \tag{5.10}$$

The first term, $dif_{speed}(t)$, measures the difference between speeds of velocities in $r(t)$ and $r_{new}(t)$:

$$dif_{speed}(t) = Speed(\Delta S(t)) - Speed(\Delta S_{new}(t)) \tag{5.11}$$

where $Speed(\Delta S(t))$ is defined in Equation 5.7. $Speed(\Delta S_{new}(t))$, based on $r_{new}(t)$, is constructed in the same manner.

The second term in Equation 5.10 accounts for the difference between directions of velocities in $r(t)$ and $r_{new}(t)$.

$$dif_{direction}(t) = \|\vec{\gamma}(t) - \vec{\gamma}_{new}(t)\| \cdot \lambda_{direction} \tag{5.12}$$

where $\lambda_{direction}$ is a constant penalty. $dif_{direction}(t)$ generates a cost value proportional to the measurement of direction differences between the velocities of $r(t)$ and $r_{new}(t)$. If the velocities coincide in all directions, $dif_{direction}(t)$ is zero.

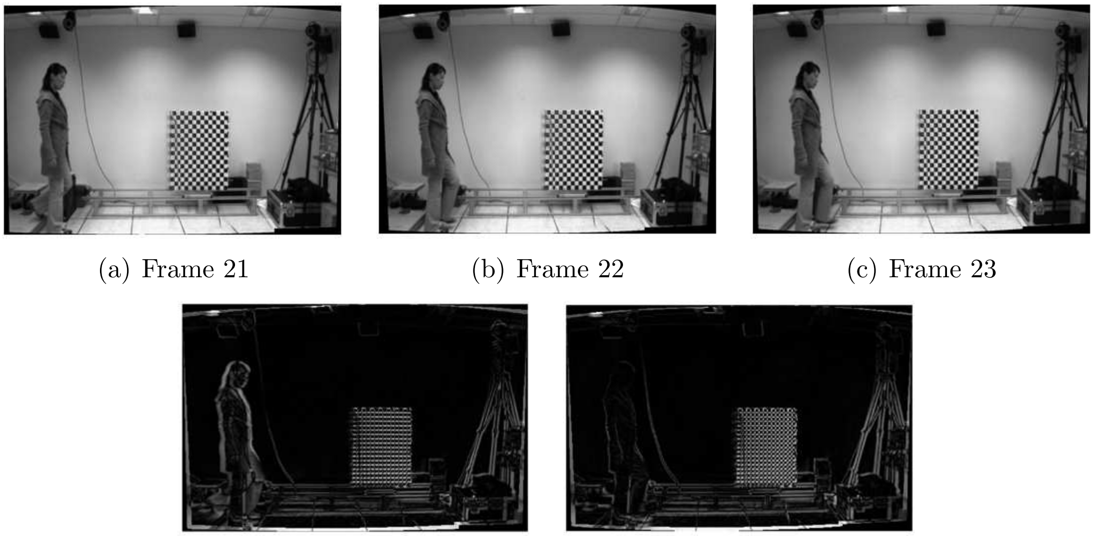
Without loss of generality, we test all the depth candidates starting from frame t, immediately prior to the entry of a foreground object into the overlapping region, where reliable depth estimates are already available. The value that provides the best match of motion velocity between $(S_{new}(t+1), S_{new}(t))$ and $(S(t+1), S(t))$, and consequently, minimizes the difference of velocity in Equation 5.10 is selected. We then apply that same calculation to estimate the foreground mosaic at time $(t-1)$, and continue until the first frame of the sequence is reached. In this manner, the foreground mosaic video in R_{non} is obtained over the time domain $[0, t]$. A simple merging of the background mosaic, as in the example of Figure 5–2f, with the foreground mosaic, results in the final dynamic mosaic video.

5.6 Experimental Results

The depth-based dynamic mosaicing method was tested on our own data. We generated video sequences of a single moving object using two input cameras with fixed parameters located along a wide baseline. Since we have not yet solved the problem of synchronization of multiple inputs through post-processing, the input cameras must be frame-synchronized during video acquisition. Camera calibration was performed using Zhang's [66] method, as implemented by Bouguet [7], and the radial lens distortion was removed after calibration. Although the sample results provided here used only two fixed video cameras as inputs, it would be a straightforward extension to take advantage of additional cameras, even with a more complex arrangement than the translationally dominant configuration used here.

To the best of our knowledge, the method introduced in this chapter is the first instance of a dynamic mosaicing algorithm that can cope with the issues of parallax and object motion if given sparse input samples. There is no contemporary reference against which to compare results. Autostitch is thus selected as a benchmark, given that it is a top-referenced representative of conventional mosaicing techniques.

As predicted in Section 5.1, when given moving objects in the scene, Autostitch generates dynamic mosaic results exhibiting jitter effects. Figure 5–7 illustrates the difference images between pairs of successive mosaicing frames. These differences arise from non-rigid movements of dynamic objects, and certain contents in



(a) Frame 21

(b) Frame 22

(c) Frame 23

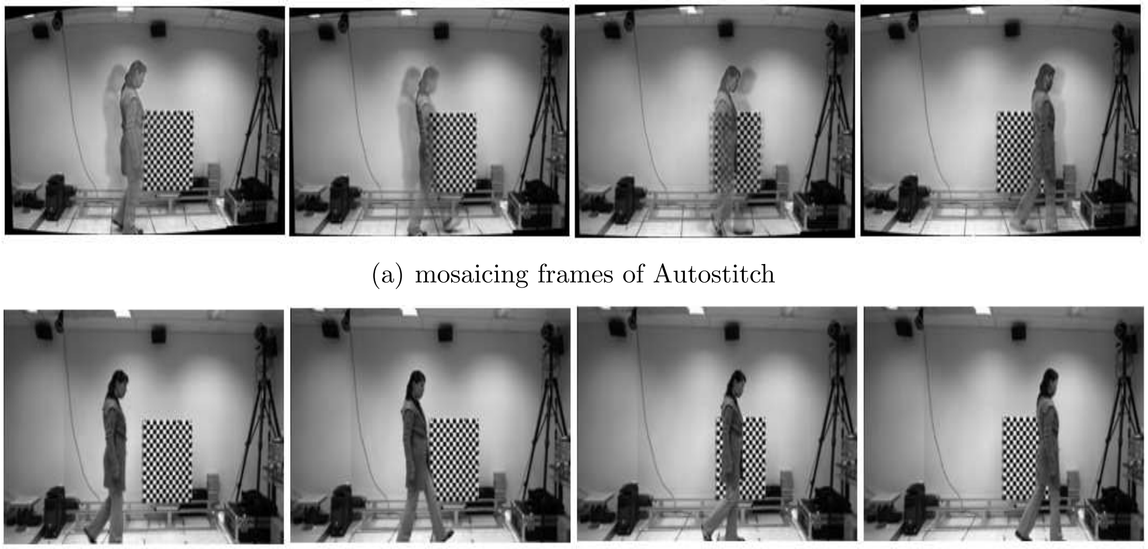
(d) $|I_{22} - I_{21}|$

(e) $|I_{23} - I_{22}|$

Figure 5–7: Illustration of jitter in the dynamic mosaic results of Autostitch. (a)-(c) Three successive mosaicing frames by Autostitch. The difference images, (d) and (e), exhibit changes of contents not only caused by the object motion but also by the jitter of static background regions.

the static background regions, due to inconsistent image registration between neighboring frames. The latter results in jitter seen on a frame-by-frame basis in the Autostitch dynamic mosaic. In contrast, our algorithm applies a common mosaicing background, as illustrated in Figure 5–2f, in each frame of the dynamic mosaic. In such a manner, it avoids the problems of jitter entirely and thus results in a consistent background throughout the entire dynamic mosaic video, equivalent to that as seen in the input videos. This is a desired property summarized in Section 1.1.

More importantly, parallax effects in the inputs lead to ghost errors observed in mosaicing frames of Autostitch, as illustrated in Figure 5–8a. As summarized in Chapter 2, parallax is also a challenge to conventional dynamic mosaicing algorithms.



(a) mosaicing frames of Autostitch

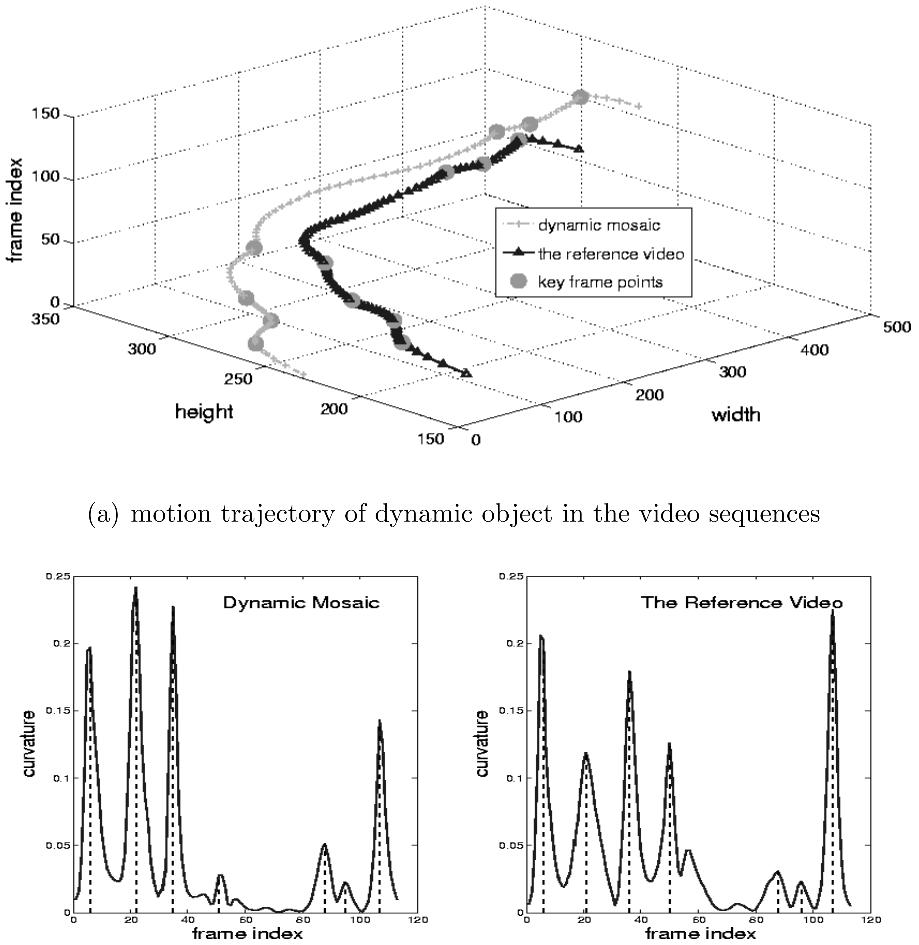
(b) mosaicing frames of our method

Figure 5–8: (a) Illustration of ghost errors in the dynamic mosaic result of Autostitch. (b) The corresponding frames built by our algorithm are free of such errors.

Although their improved image registration techniques may avoid jitter problems by reducing the impact of dynamic objects in the scene, they cannot overcome the parallax issue, which remains evident even in the static environment. The results of Figure 5–8b illustrate that our approach generates results free of these errors.

In order to validate the results of our algorithm with respect to preservation of motion consistency in the spatiotemporal domain, we compare these to the output of a reference camera, which was located well behind the baseline between two input cameras. This allows us to obtain a similar FOV as the virtual mosaicing camera. As illustrated in Figure 5–9a, the video acquired by the reference camera and the synthesized dynamic mosaicing video present similar motion trajectories, with the same sequence of instants, which are identified as maxima in the curvature of trajectory, as shown in Figure 5–9b. The corresponding frame indices of these maxima are also marked on the trajectories as key frame points. The similarity of trajectories and instants between the reference video and the dynamic mosaic confirms that the latter preserves not only motion consistency in the spatiotemporal domain but also provides an identical perception of motion. Samples from both the reference video and the dynamic mosaic, taken from identical frames indices, are provided in Figure 5–10 to illustrate motion consistency.

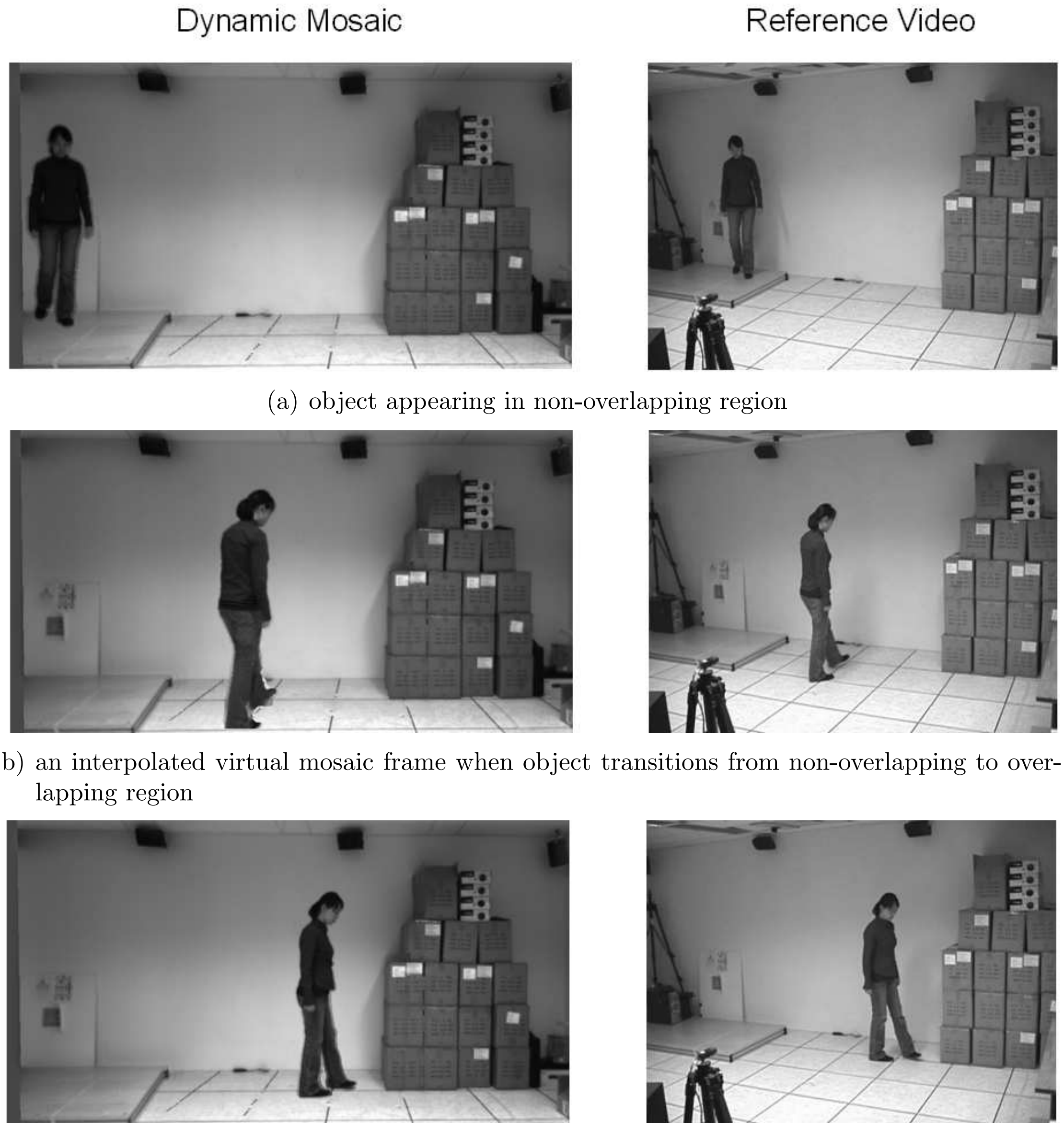
The current depth-based dynamic mosaicing algorithm presents a successful processing pipeline that generates reasonable mosaic video containing a single moving object. We have not yet addressed more complicated motion patterns, such as those involving multiple objects with occlusion and disocclusion effects. In such cases, moving objects are unfortunately detected as a single silhouette when they overlap, as shown in Figure 5–11a. This common silhouette is used to calculate the depth estimate for each separately moving object, based on the respective motion trajectory analysis. As shown in Figure 5–11(b-c), at time instant t, the entire content of the silhouette will be warped onto a different position (I or II) of the output mosaicing frame depending on which object, i.e., the male or female subject, is used to determine its depth. This results in duplicated foreground content, as appears in the output mosaic frame of Figure 5–11d. A possible solution may be to take advantage of motion tracking or 3D models of foreground objects, obtained prior to the algorithm. Ultimately, this may enable the construction of dynamic mosaics that include arbitrary motion of multiple objects.



(a) motion trajectory of dynamic object in the video sequences

(b) curvature values of the motion trajectory

Figure 5–9: (a) Motion trajectories of the reference video and the dynamic mosaic video. (b) Corresponding curvature values. Strong coherence of both the trajectories and the sequence of instants is observed between the two paths.

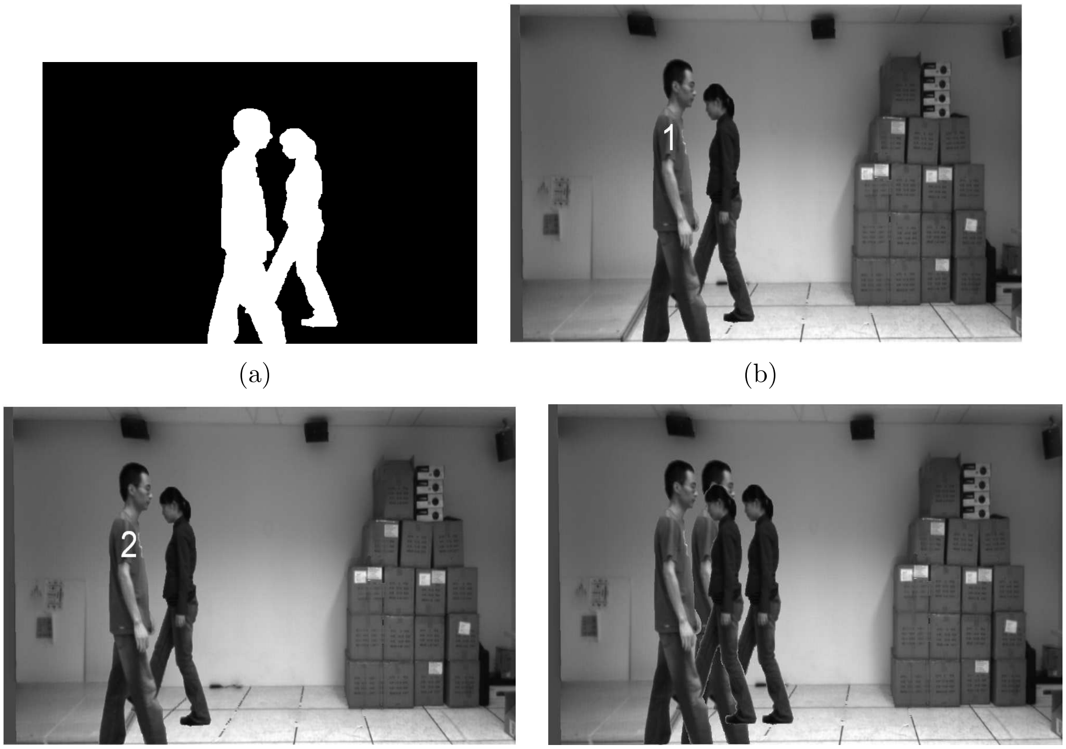


(a) object appearing in non-overlapping region

(b) an interpolated virtual mosaic frame when object transitions from non-overlapping to overlapping region

(c) object appearing in overlapping region

Figure 5–10: Comparison of frames from the dynamic mosaic to those from the reference video.



(a)

(b)

(c)

(d)

Figure 5–11: (a) The common foreground mask (silhouette) containing multiple moving objects. (b) and (c) The different warping positions of the foreground contents based on the depth estimates calculated according to motion trajectories of different moving objects. (d) The resulting duplicated foreground content in the output mosaic frame.

Another limitation of the current dynamic mosaicing approach is that it must first obtain reliable depth estimates of the foreground layer, obtained from the overlapping region, in order to propagate these into the neighboring non-overlapping regions. As a consequence, the algorithm is presently unsuitable for on-line applications, unless all moving foreground objects are first seen in the overlapping regions. Overcoming this limitation remains a problem for future work.

The shadow of the moving objects is unfortunately removed in the current dynamic mosaicing results. This is due to our training of the statistical background model, which regards such shadows as part of the static background. In order to preserve such shadows and render them properly in the dynamic mosaicing results, a shadow and light source detection algorithm must be included in the implementation.

5.7 Summary

Chapter 4 introduced our depth-based image mosaicing algorithm and demonstrated its application to static scenes. This chapter described an extension to the algorithm as necessary for generating dynamic mosaics.

Following the foreground-background segmentation, we project the separated layers onto the mosaicing image plane according to their depth estimates. This results in our construction of the first reasonable chronologically consistent dynamic mosaic video, which includes contents from both overlapping and non-overlapping regions and preserves spatiotemporal motion consistency, even when presented with challenging inputs exhibiting obvious parallax effects.

The new algorithm provides an efficient pipeline for construction of dynamic mosaics, which overcomes both the challenges of parallax and dynamic objects. We believe the work presented in this chapter will lead to many promising applications of dynamic video mosaicing.

CHAPTER 6
Conclusions

6.1 Future Directions

Dynamic Mosaic of Multiple Objects

As discussed in Section 5.6, although the current depth-based dynamic mosaicing algorithm accommodates the motion pattern of a single moving object, it is not designed to cope with more complicated motions of multiple objects in the scene.

The first problem that must be tackled in this case is the difficulty of background segmentation. When the dynamic foreground contents occupy a significant portion of the FOV, it is hard to distinguish them from the fixed background based on statistical information. Thus, a robust segmentation method is required to handle such conditions. Secondly, the complex motion pattern of group activity, including multiple independent trajectories with occlusion and disocclusion effects make it difficult to preserve motion consistency in the output dynamic mosaic. Access to motion tracking information or 3D models of foreground objects may prove helpful in this case.

Sensor Fusion for Improved Segmentation

Color segmentation, foreground-background segmentation and depth estimation are three correlated factors related to depth discontinuities, so that the combination of them in the form of sensor fusion may improve their overall results.

In the thesis, the quality of depth-based image mosaicing results is in large part dependent on the quality of color segmentation. A careful parameter tuning procedure is required to achieve the appropriate size of color segments that preserve edges well along depth discontinuities. If integrated the foreground-background segmentation and depth estimation, it should be possible to design a fully automatic color segmentation algorithm.

Evaluation Metric

A widely acknowledged shortcoming in the field of image mosaicing is the lack of an evaluation method to compare algorithms quantitatively. Because of the problematic issues involved in producing ground truth results, and the challenge of fairly comparing the results obtained by different synthesis methods, e.g., multiperspective vs. single perspective projection, mosaic quality is invariably judged subjectively. It would be helpful to create a novel gradient domain analysis to determine the location and the severity of mosaicing errors.

A Real-time Application

For our target application of a video-conferencing system, optimization of these algorithms for video rate performance at high resolution is vital. Possible ideas we would like to explore, include exploitation of the parallel computation abilities of a GPU to produce fast rendering for overlapping regions, and the use of pre-calibrated stereo cameras or laser scanners to construct an accurate depth map of the entire scene. The latter would reduce the computation required to obtain depth estimates.

Efficient Video Compression and Transmission

Video contains redundant information, such as static scene content that appears repeatedly in consecutive frames. Following the approach of the majority of video compression algorithms, representations that preserve significant structures and motion can improve the efficiency of transmission, browsing and searching of video sequences. Our novel depth-based dynamic mosaicing technique eliminates much of the redundancy of video sequences and generates three related outputs: the static background panorama, the dynamic foreground layers, and their corresponding depth estimates. With further research, it may be possible to reconstruct the individual input video stream from these three related representations, thus leading to advances in video compression, transmission and indexing method.

6.2 Concluding Remarks

This thesis addressed the problem of overcoming the challenges of parallax and motion in image mosaicing. It removed the traditional constraints of parallax-free camera motion or dense sampling of the environment and contributed to the development of a dynamic mosaicing algorithm that copes with non-parallax-free video inputs.

A novel image mosaicing algorithm built upon these techniques, i.e., the integration of image-based rendering and optimization algorithms, was introduced and validated in Chapter 3. However, its computational cost motivated the investigation of more efficient methods.

Another algorithm, which formulated the image mosaicing as a depth-based view synthesis problem, and notably included the contents in non-overlapping regions between sources by using a depth propagation procedure, is discussed in Chapter 4. As demonstrated in Chapter 5, the depth-based image mosaicing algorithm, if integrated with the technique of foreground-background segmentation and the consistent motion perception criterion, can be extended to generate reasonable dynamic mosaics that are robust to both parallax and motion issues.

Conventional image mosaicing techniques warp inputs by estimating the geometric relationship, i.e., the camera motion models explicitly, so that input images are aligned particularly in their regions of overlap. However, by incorporating depth cues, our new approaches applied the smooth appearance connection criterion to ensure natural transition between contents in panoramas of static scenes. Similarly, they also imposed a consistent motion perception criterion with respect to moving objects. This preserved continuity of object movements in dynamic mosaicing videos. Experiments demonstrated that the new algorithms generate promising results, even when given challenging inputs on which traditional image mosaicing algorithms tend to fail. Nevertheless, it bears comment that the use of depth cues to overcome the issues of parallax and object motion also imposes constraints. Specifically, we assume that the inputs present observable disparity differences; our algorithms would fail if presented depth-free inputs, which do not contain sufficient stereo information to enable further depth estimation. Furthermore, unlike traditional image mosaicing algorithms, our approach requires camera calibration, which may be a time-consuming task.

Finally, our investigation of techniques to overcome parallax and motion issues has exposed opportunities for further improvements, and also raised further open questions.

References

[1] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski. Photographing long scenes with multi-viewpoint panoramas. In *SIGGRAPH'06: Proceedings of the 33th annual conference on Computer graphics and interactive techniques*, pages 853–861. ACM, 2006.

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

[3] A. Agarwala, K.C. Zheng, C. Pal, M. Agrawala, M. Cohen, B. Curless, D. Salesin, and R. Szeliski. Panoramic video textures. In *SIGGRAPH'05: Proceedings of the 32th annual conference on Computer graphics and interactive techniques*, pages 821–827. ACM, 2005.

[4] R.A. Alex, Y. Pritch, D. Lischinski, and S. Peleg. Dynamosaicing: Mosaicing of dynamic scenes. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29:1789–1801, 2007.

[5] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework: Part 1. Technical Report CMU-RI-TR-02-16, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2002.

[6] M. Bartoli, N. Dalal, B. Bose, and R. Horaud. From video sequence to motion panoramas. In *MOTION'02: Proceedings of the Workshop on Motion and Video Computing*, pages 201–207. IEEE, 2002.

[7] J.V. Bouguet. Camera calibration toolbox for Matlab, 2003. `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

[8] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26:1124–1137, 2004.

[9] Y. Boykov, R. Zabih, and S. Gortler. Multi-camera scene reconstruction via graph cuts. In *ECCV'02: Proceedings of the 7th European Conference on Computer Vision-Part III*, pages 82–96, 2002.

[10] G. Bradski and J. Davis. Motion segmentation and pose recognition with motion history gradients. *Machine Vision and Applications*, 13(3):174–184, 2002.

[11] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR'00: Proceedings of the 2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 690–696, 1999.

[12] M. Brown and D.G. Lowe. Autostitch:: a new dimension in automatic image stitching. `http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html`.

[13] M. Brown and D.G. Lowe. Recognising panoramas. In *ICCV'03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, pages 1218–1225, 2003.

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

[15] S.C. Chen. Quicktime VR: An image-based approach to virtual environment navigation. In *SIGGRAPH'95: Proceedings of the 22th annual conference on Computer graphics and interactive techniques*, pages 29–38. ACM, 1995.

[16] R. Collins. A space-sweep approach to true multi-image matching. In *CVPR'96: Proceedings of the 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 358–363, 1996.

[17] P. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH'97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 369–378, New York, NY, USA, 1997.

[18] A. Del Bue, F. Smeraldia, and L. Agapitoa. Non-rigid structure from motion using ranklet-based tracking and non-linear optimization. *Image and Vision Computing*, 25(3):297–310, 2007.

[19] E.W. Dijkstra. A note on two problems in connection with graphs. *Numerische Math*, pages 262–271, 1959.

[20] A. Eden, M. Uyttendaele, and R. Szeliski. Seamless image stitching of scenes with large motions and exposure differences. In *CVPR'06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2498–2505, 2006.

[21] B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions: A texture classification example. In *ICCV'03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, pages 456–463, 2003.

[22] R. Gupta and R.I. Hartley. Linear pushbroom cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(9):963–975, 1997.

[23] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, New York, NY, USA, 2000.

[24] C.T. Hsu and Y.C. Tsan. Mosaics of video sequences with moving objects. In *International Conference on Image Processing (ICIP)*, pages 387–390, 2001.

[25] M. Irani, P. Anandan, and S. Hus. Mosaic based representation of video sequence and their applications. In *ICCV'95: Proceedings of the Fifth IEEE International Conference on Computer Vision*, pages 605–611, 1995.

[26] Y. Ivanov, A. Bobick, and J. Liu. Fast lighting independent background subtraction. *International Journal of Comput. Vision*, 37(2):199–207, 2000.

[27] S. Iwase and H. Saito. Parallel tracking of all soccer players by integrating detected positions in multiple view images. In *ICPR'04: Proceedings of the Pattern Recognition, 17th International Conference on*, pages 751–754, 2004.

[28] R.J. Jagacinski, W.W. Johnson, and R.A. Miller. Quantifying the cognitive trajectories of extrapolated movements. *Journal of Exp. Psychology: Human Perception and Performance*, 9:43–57, 1983.

[29] J. Jia and C.K. Tang. Image stitching using structure deformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(4):617–631, 2008.

[30] Y. Kameda and M. Minoh. A human motion estimation method using 3-successive video frames. In *VSMM'96: In Proceeding of the International Conference on Visual Systems and Multimedia*, pages 135–140, 1996.

[31] F. Kristensen, P. Nilsson, and V. Owall. Background segmentation beyond RGB. In *ACCV'06: the 7th Asian Conference on Computer Vision*, pages 602–612, 2006.

[32] D.S. Lee. Effective Gaussian mixture learning for video background subtraction. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27:827–832, 2005.

[33] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. In *ECCV'04: Proceedings of the 8th European Conference on Computer Vision*, pages 377–389, 2004.

[34] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH'96: Proceedings of the 23th annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996.

[35] S.N. Lim, A. Mittal, L. Davis, and N. Paragios. Fast illumination-invariant background subtraction using two views: Error analysis, sensor placement and applications. In *CVPR'05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1071–1078, 2005.

[36] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[37] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(10):1615–1630, 2005.

[38] D. Newtson and G. Engquist. The perceptual organization of ongoing behavior. *Journal of Experimental Social Psychology*, 12:436–450, 1976.

[39] S. Peleg and J. Herman. Panoramic mosaics by manifold projection. In *CVPR'97: Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 338–343, 1997.

[40] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet. Mosaicing on adaptive manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(10):1144–1154, 2000.

[41] Z. Qi and J.R. Cooperstock. Automated change detection in an undersea environment using a statistical background model. In *Oceans 2007*, pages 1–6, 2007.

[42] Z. Qi and J.R. Cooperstock. Overcoming parallax and sampling density issues in image mosaicing of non-planar scenes. In *British Machine Vision Conference (BMVC)*, pages 199–207, 2007.

[43] Z. Qi and J.R. Cooperstock. Depth-based image mosaicing for both static and dynamic scenes. In *ICPR'08: Proceedings of the 19 th International Conference on Pattern Recognition*, 2008.

[44] P. Rademacher and G. Bishop. Multiple-center-of-projection images. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 199–206, 1998.

[45] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50(2):203–226, 2002.

[46] H.S. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18:814–830, 1996.

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

[48] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *CVPR'03: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 195–202. IEEE Computer Society, June 2003.

[49] S.M. Seitz and J. Kim. The space of all stereo images. *International Journal of Computer Vision*, 48(1):21–38, 2002.

[50] J. Shade, S. Gortler, L.W. He, and R. Szeliski. Layered depth images. In *SIGGRAPH'98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 231–242, 1998.

[51] H.Y. Shum, S.C. Chan, and S.B. Kang. *Image-Based Rendering*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[52] H.Y. Shum and R. Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *ICCV'98: Proceedings of the Sixth International Conference on Computer Vision*, pages 953–960, 1998.

[53] H. Sidenbladh. Detecting human motion with support vector machines. In *ICPR'04: Proceedings of the Pattern Recognition, 17th International Conference on*, pages 188–191, 2004.

[54] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR '99: Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 23–25, 1999.

[55] C. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537, 1999.

[56] R. Szeliski and H.Y. Shum. Creating full view panoramic image mosaics and environment maps. In *SIGGRAPH'97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 251–258, 1997.

[57] H. Tao, S.H. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *ICCV'01: Proceedings of the Eighth IEEE International Conference on Computer Vision*, pages 532–547, 2001.

[58] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[59] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(5):878–892, 2008.

[60] M. Uyttendaele, A. Eden, and R. Szeliski. Eliminating ghosting and exposure artifacts in image mosaics. In *CVPR'01: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 509–516, 2001.

[61] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV'03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 734, 2003.

[62] Y. Wexler and D. Simakov. Space-time scene manifolds. In *ICCV'05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 858–863, 2005.

[63] R. Yang, M. Pollefeys, H. Yang, and G. Welch. A unified approach to real-time multi-resolution multi-baseline 2d view synthesis and 3d depth estimation using commodity graphics hardware. *International Journal of Image and Graphics*, 4(4):627–651, 2004.

[64] J.M. Zacks and B. Tversky. Event structure in perception and conception. *Psychological Bulletin*, 127:3–21, 2001.

[65] Y. Zhang and C. Kambhamettu. Stereo matching with segmentation-based cooperation. In *ECCV'02: Proceedings of the 7th European Conference on Computer Vision-Part II*, pages 556–571, 2002.

[66] Z.Y. Zhang. A flexible new technique for camera calibration. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22:1330–1334, 2000.

[67] K. Zheng, S.B. Kang, M. Cohen, and R. Szeliski. Layered depth panoramas. In *CVPR'07: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[68] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: The crossed-slits projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(6):741–754, 2003.