SCENE REPRESENTATION AND VIEW SYNTHESIS IN IMAGE-BASED RENDERING

Xiaoyong Sun

A Thesis submitted to the University of Ottawa in partial fulfillment
of the requirements for the degree of Doctor of Philosophy in
Electrical Engineering

December 2007

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, Canada

In memory of my father ...

Contents

List of Figures

Abstract

Image-Based Rendering is a technology that can be used to develop systems for navigation in real-image-based virtual environments, potentially providing a high-quality realistic experience. Scene representation and view synthesis are major research topics in this field and form the main focus of this thesis. Three major topics investigated in detail are high-quality panorama generation, view interpolation and a simplified Concentric Mosaics technique. First, a novel optimization model is proposed for registration of the overlap-area between two adjacent images taken by a camera mounted and rotated on a tripod, and a new algorithm is given to significantly reduce the accumulated errors when stitching multiple images to generate 360° panoramas. The algorithms have been implemented based on matching features to significantly reduce the computations. Second, a matching-feature-based view interpolation algorithm is proposed and the triangulation of the images, combined with an affine transformation model, has been applied for the texture mapping. In addition, a novel disparity estimation algorithm is studied for view interpolation. Special transformations determined from the physical imaging conditions to minimize the difference between two source views are used in the algorithm. Finally, a simplified implementation of the Concentric Mosaics technique is proposed. The camera positions, where the pre-captured images are taken, are estimated from the pre-captured images. The mathematical equations and the optimal solution of such large scale linear equations with noisy coefficients are given, together with a set of associated methods such as a closed-loop constraint, a ratio fitting technique, and an angle grouping method to improve the estimation precision. Simulation results demonstrate that each proposed view synthesis method can generate valid views of good quality and satisfy Image-Based Rendering application requirements.

Acknowledgements

First and foremost, I would like to express my heartfelt gratitude towards my supervisor Dr. Eric Dubois for giving me this opportunity to work with him. I did benefit not only from his helpful guidance and motivation at all times whenever I needed them, but am still benefiting from his character which leads me on to become an excellent researcher like him.

I also want to thank my mother whose support over the years and her encouragement has made me self-confident to face life's challenges. Also thanks to the help from all my friends and appreciate the help from my colleagues in the VIVA lab.

Also I must thank the National Capital Institute of Telecommunications Canada, the Learning Objects Repositories Network (LORNET), and Natural Science and Engineering Research Council of Canada (NSERC), which provided the funding for this research work.

Chapter 1

Introduction

Navigation in a virtual environment providing a high-quality realistic experience has been a very active research topic in recent years. Although most of the currently-used techniques are based on computer-graphics methods to simulate a photo-realistic effect, the techniques of Image-Based Rendering [1], [2], [3], [4], [5], [6] bring potential to generate truly photo-realistic views based on real images.

The idea of Image-Based Rendering (IBR) is to generate arbitrary novel views of a scene, anywhere within a certain navigation area and in any viewing direction, based on a set of pre-captured images. It has become an active research topic with the availability of increased computing power and network bandwidth. The idea is illustrated in Fig. 1.1. The panoramic view at the top of Fig. 1.1 represents a real scene (a real 3D environment). After sufficiently many pre-captured images, taken by an ordinary camera at different positions and in different directions, are obtained and stored, a view at any arbitrary position and in any arbitrary direction can be synthesized from the pre-captured-image database. These synthesized arbitrary views can be regarded as have been taken by a virtual camera. In this way, a real-image-based virtual environment is constructed and a user can navigate in it by controlling the trajectory and orientation of the virtual camera, using either a local or a remote terminal. IBR can provide the user with a highly realistic experience in many applications such as e-commerce, teleconferencing, view-based maps, virtual museum visiting, new-worker

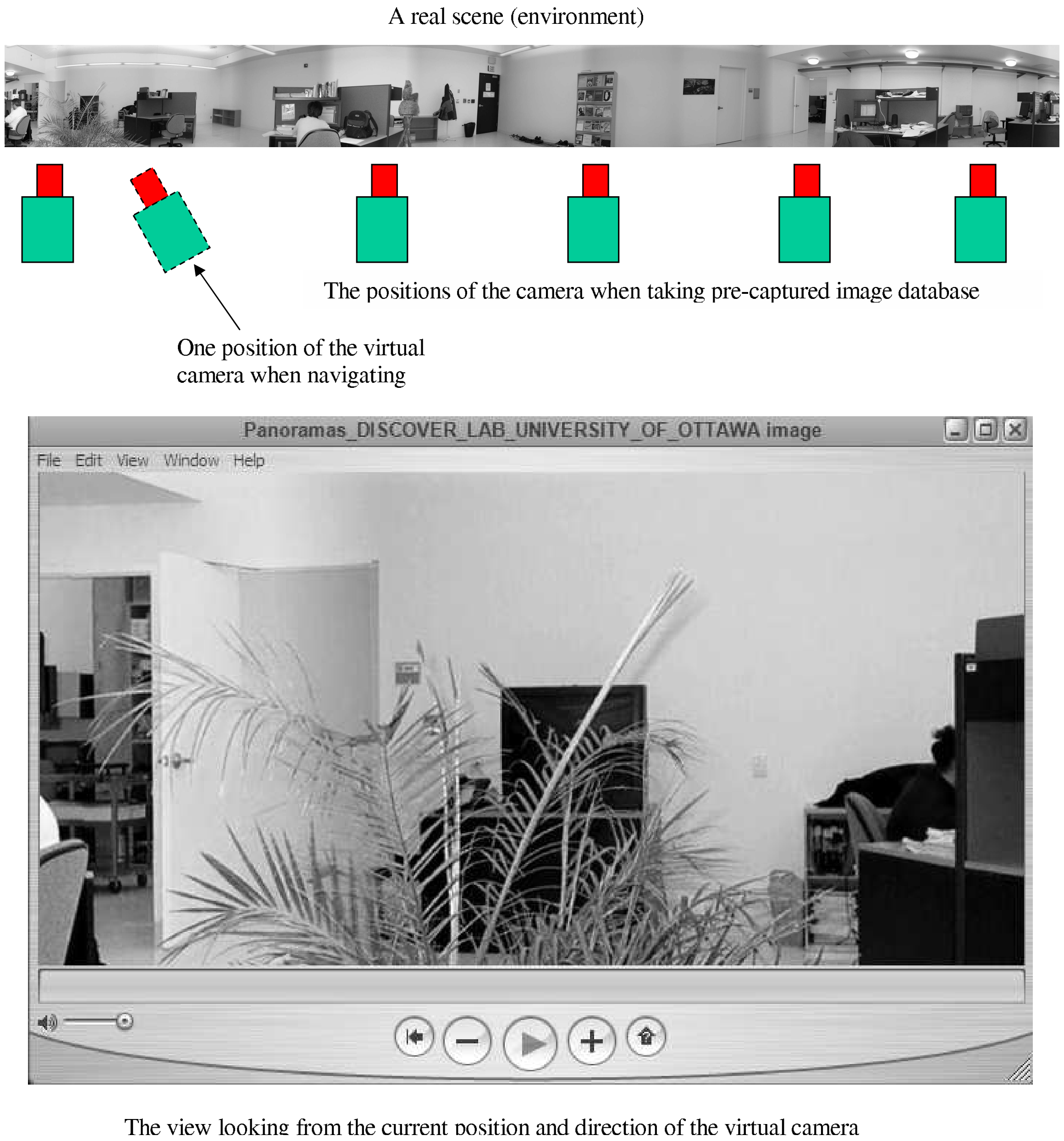

A real scene (environment)
The positions of the camera when taking pre-captured image database
One position of the virtual camera when navigating
Panoramas_DISCOVER_LAB_UNIVERSITY_OF_OTTAWA image
File
Edit
View
Window
Help
The view looking from the current position and direction of the virtual camera

Figure 1.1: The illustration of Image-Based rendering

training, e-education, etc.

However, the implementation of an IBR system is very difficult. The research is still in an initial stage and much remains to be done [7]. The current research work is focusing on topics such as

- How many pre-captured images are necessary and sufficient to represent a particular environment?
- How to determine the camera positions and shooting directions in order to obtain such a set of necessary and sufficient pre-captured images, if more than one set exists?
- What are efficient acquisition methods to obtain such sets, or at least one set, of necessary and sufficient pre-captured images?
- Given one such set of pre-captured images, how can the novel views be synthesized (with good quality)?
- What are the optimal IBR techniques in the sense of increasing acquisition efficiency, reducing rendering complexity and improving rendering quality?
- How can these pre-captured images, which usually involve a huge amount of data, be compressed and stored so that they can be accessed efficiently during the real-time rendering?

There exists neither a general answer to any of these questions nor a general method for all IBR applications. Current work is focusing on different specific methods for different applications. Many IBR approaches have been proposed for specific applications. In the different IBR approaches, the methods to obtain the pre-captured images and to synthesize novel views are very different and are the distinguishing features for different approaches. The methods of view synthesis usually depend on the data structures of the pre-captured images, which are motivated by the different strategies for scene representation and determined by the acquisition methods.

Each IBR method has its own advantages and disadvantages. In the future, a complete IBR system might be hybrid [8] and compatible with different structures of the pre-captured image data obtained from different methods, using concepts such as plenoptic primitives [9].

Neither a general answer to any of the above questions nor a general method for IBR applications will be given in this thesis. Only after each particular technique has been thoroughly studied can the above questions be precisely answered. In this thesis, different approaches for view synthesis from the pre-captured images will be studied. The methods to acquire pre-captured images and to organize these pre-captured images in the different approaches will be investigated for generating novel views from the pre-captured images. We will find that view synthesis is implemented through the plenoptic function, which is used to represent the scene and relates the synthesized views to the pre-captured images. In order to associate the pre-captured images with the plenoptic function, the camera positions associated with these pre-captured images must be known.

In the following section, the current research on different approaches for IBR will be reviewed and organized in a general framework. Then, the different approaches will be compared with each other, and the conclusions will determine the thesis orientation. The contributions will be described, followed by an overview of the organization of the thesis.

1.1 Overview of current research on IBR

The framework of IBR is based on the concept of the continuous plenoptic function [10]. In various IBR techniques, the environments are represented by pre-captured images that contain discrete samples of the plenoptic functions. Based on the reconstructed plenoptic functions, novel views can be synthesized for any specified position and viewing direction.

Each view is constructed from all light rays entering the virtual camera, passing through the projection center of the camera at every possible angle (θ, ϕ) within the field of view (FOV), where θ is along the horizontal direction and ϕ is along the vertical direction. The entire set of light rays that can be perceived at every possible location (V_x, V_y, V_z) and every time t can be represented by a seven-dimensional function, if each light ray is decomposed into different wavelengths λ, as

$$P(\theta, \phi, \lambda, t, V_x, V_y, V_z), \tag{1.1}$$

where P is the spectral radiance. The seven-dimensional plenoptic function can be reduced to six dimensions by ignoring the time variable, which is appropriate for static environments. The plenoptic function can further be reduced to five dimensions by eliminating the wavelength variable,

$$P_i(\theta, \phi, V_x, V_y, V_z) = \int P(\theta, \phi, \lambda, V_x, V_y, V_z) q_i(\lambda) d\lambda, \ i = 1, 2, 3, \tag{1.2}$$

where the $q_i(\lambda)$ are ideally color-matching functions of human vision. In practice, they are sensitivities of three color sensors, such as $r(\lambda)$, $g(\lambda)$ and $b(\lambda)$. Thus, each light ray will consist of three components for the tri-chromatic representation of a color view in a given color space. We use vector P to denote $[P_1, P_2, P_3]$ for the color-space representation of light rays,

$$P(\theta, \phi, V_x, V_y, V_z). \tag{1.3}$$

Although it is difficult, if not impossible, to capture all the light rays within a certain spatial area, the plenoptic function provides a mathematical model for the scene representation where the light rays are organized in the viewer's coordinate system.

The objective of IBR is to generate arbitrary views within certain ranges of viewing positions and directions by obtaining or reconstructing the plenoptic functions from the discrete plenoptic samples, which are extracted from the pre-captured images. The plenoptic function is theoretically a continuous function, but it is represented in discrete format in practice and its value at any arbitrary position and direction can be obtained through interpolation. The ranges of possible viewing positions and directions define the navigation space.

In [11], the different image-based representations have been classified based on whether or not geometric information about the scene is required and what kind of geometric information (implicit or explicit) is required for a particular technique. However, it will be helpful for any rendering algorithm if the scene geometry is available. In order to explore the relationship between different methods and to illustrate the application scenarios of each method, the different techniques that have been developed can be associated with one of three main approaches as shown in Fig. 1.2.

One goal of IBR is to eliminate the tedious work involved in building 3D model-based scene representations using computer graphics primitives. If the 3D objects in computer graphics can be generated using real images, the procedure of building 3D models may be largely simplified. In addition, the previous rendering framework can still be used with improved rendering quality due to the more realistic generated views. This form of representation is similar to the idea of representation using source descriptions in [12]. Because the coordinate systems for the source descriptions are world systems, the plenoptic functions are not explicitly expressed in this approach and the rendering algorithms are similar to those in computer graphics.

Recently, research has been carried out on methods to generate novel views from the pre-captured images in scenarios where the camera positions from where the pre-captured are taken, and the virtual camera positions from where the novel views are to be obtained, are close to each other. This approach is called *general view interpolation* here and the methods developed include view interpolation [1], view morphing [13], view transfer [14], [15], [16], etc. Only the reference views taken at positions that are near the virtual cameras are required and only novel views with the virtual camera close to the positions where several (at least two) pre-captured images have been taken can be generated. In such a scenario, a few pre-captured images might be sufficient to generate novel views within some limited local areas and toward some specified viewing directions. This may be adequate for some applications. Moreover, when more pre-captured images are available, the navigation areas and the ranges of viewing directions can consequently be increased. Thus, these techniques are very

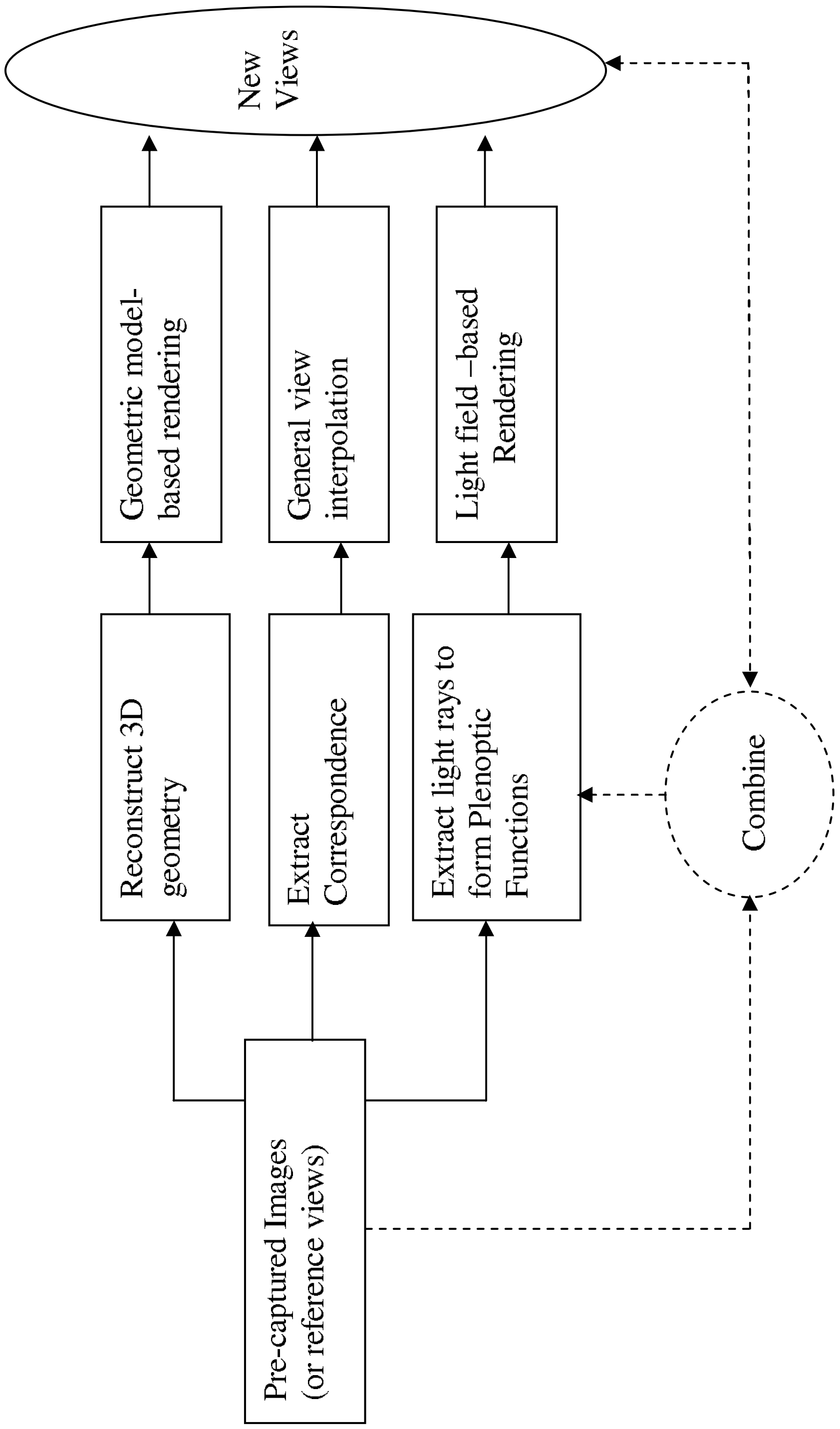


Figure 1.2: Overview of different scene representations for IBR

flexible for different application requirements.

Both of the above approaches require certain geometric information about the scene, but in different formats. In the 3D-reconstruction approach, explicit geometry of the scene has to be reconstructed for rendering, whereas implicit geometry of the scene is usually used in the general-view-interpolation approaches. The relationship between the above two approaches is illustrated in Fig. 1.3. Both methods essentially include two steps: specifying the imaging position of the 3D scene points in the novel views and mapping the texture from the reference views to the novel views. If the scene geometry can be reconstructed, either from the reference images or from any other methods such as using a range finder, the novel views can be obtained based on the projection theory using the texture from the reference images (assuming that the scene geometry and the texture in the reference images have been registered if the scene geometry is obtained from a range finder). Thus, explicit 3D models

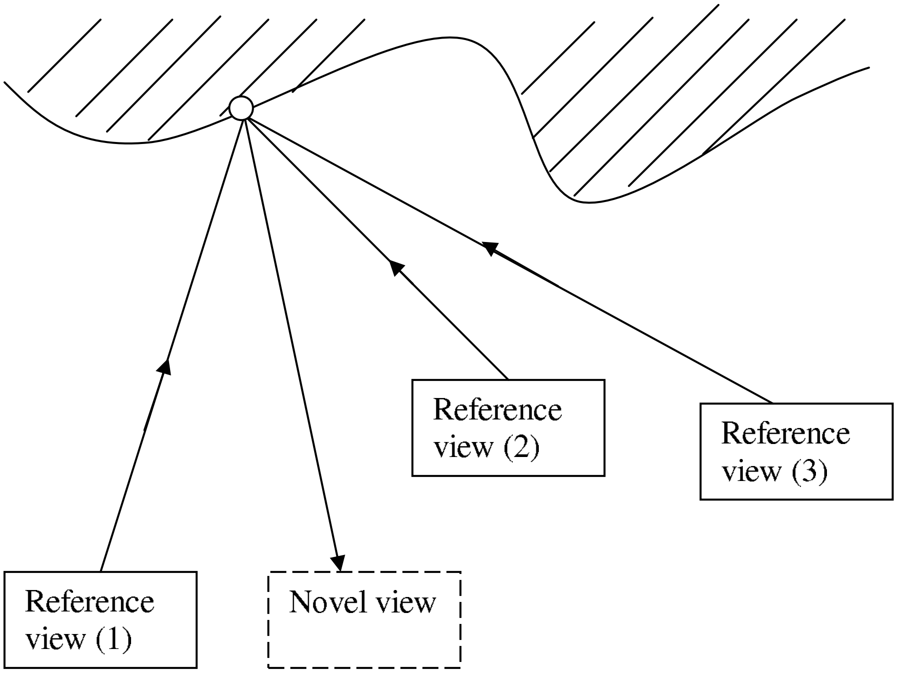


Figure 1.3: Rendering with implicit or explicit geometry

of the scene are reconstructed in this approach. In the general-view-interpolation approach, the novel views can be directly generated from the reference images using the camera relationships among the reference views, and between the reference views and the novel views. The 3D scene points are still used in order to obtain the above relationships although they may not be explicitly reconstructed in space.

The plenoptic functions can also be explicitly defined using the light-field descriptions, with pre-captured images that are taken using specially-designed procedures. The manner in which the light rays are organized, the data structures of pre-captured images, and the dimensions of the plenoptic functions can be very different with different techniques. Thus, they have different rendering algorithms, navigation area constraints, technical requirements for implementation, etc. The methods used within this category are technique dependent. For example, Light Field Rendering [4] indexes each light ray in the plenoptic function using two planes, whereas the Concentric Mosaics technique [17] uses a very different way to index each column in the pre-captured images taken on a circular path.

The dashed-line parts in Fig. 1.2 describe possible conversions between different representations. When the 3D models of the scene are reconstructed (both geometry and texture), the plenoptic functions can be obtained by moving a virtual camera to capture the required images for light-field representations. Then the model-based rendering is converted to the light-field approach. Similarly, the plenoptic functions can be obtained from the pre-captured images together with additional images that can be interpolated through the general view interpolation approach for particular ranges of viewing positions and directions. Thus the essential difference between different methods is the different data structures of the plenoptic function representations. Different approaches may be used collaboratively in one particular method. The plenoptic functions may also be converted from one format to another, or to one with reduced (or equivalent) dimensions. The considerations for selecting one particular technique are generally application-based.

The various types of image-based representations that have been developed can be summarized as shown in Fig. 1.4, which is a more detailed version of Fig. 1.2. In the following sections, we briefly introduce the major techniques within these three approaches:

- 3D geometry reconstruction (the top branch within dashed rectangle in Fig. 1.4)
- general view interpolation (the middle branch within dashed rectangle in Fig. 1.4)
- light-field description (the bottom branch within dashed rectangle in Fig. 1.4)

1.1.1 Approaches based on 3D geometry reconstruction

The methods in this category use the explicit geometry of the scene to be represented. The geometry of the scene can be obtained from a range camera, or range finder, and then be registered with the correspondent texture in the pre-captured images. However, a representation of the scene geometry is most often obtained using correspondences (matching features or dense disparity maps) between the pre-captured images along with the appropriate camera models.

After completely reconstructing the 3D geometry of the scene, IBR has been converted to the traditional model-based rendering of computer graphics, allowing the advanced techniques in computer graphics to be used in the rendering procedure. For example, the 3D warping algorithm is factorized into a simple pre-warping stage and a standard texture mapping procedure [18]. Texture mapping has been extensively studied in computer graphics and can be accelerated by standard graphics hardware.

It is well known in computer vision that the precise 3D reconstruction of an environment is very difficult and expensive. However, as long as some kind of 3D model is available (precisely, roughly, or even partially), it is helpful to use it in IBR for view synthesis. These 3D models are usually mesh or volumetric models [19]. Many methods for 3D model reconstruction have been reported in the literature, such as visual-hull-based methods [20], [21], and voxel-based methods[22], etc.

Realizing the difficulties to precisely reconstruct the environment with a perfect 3D geometric model, the locally reconstructed model using the stereo technique has

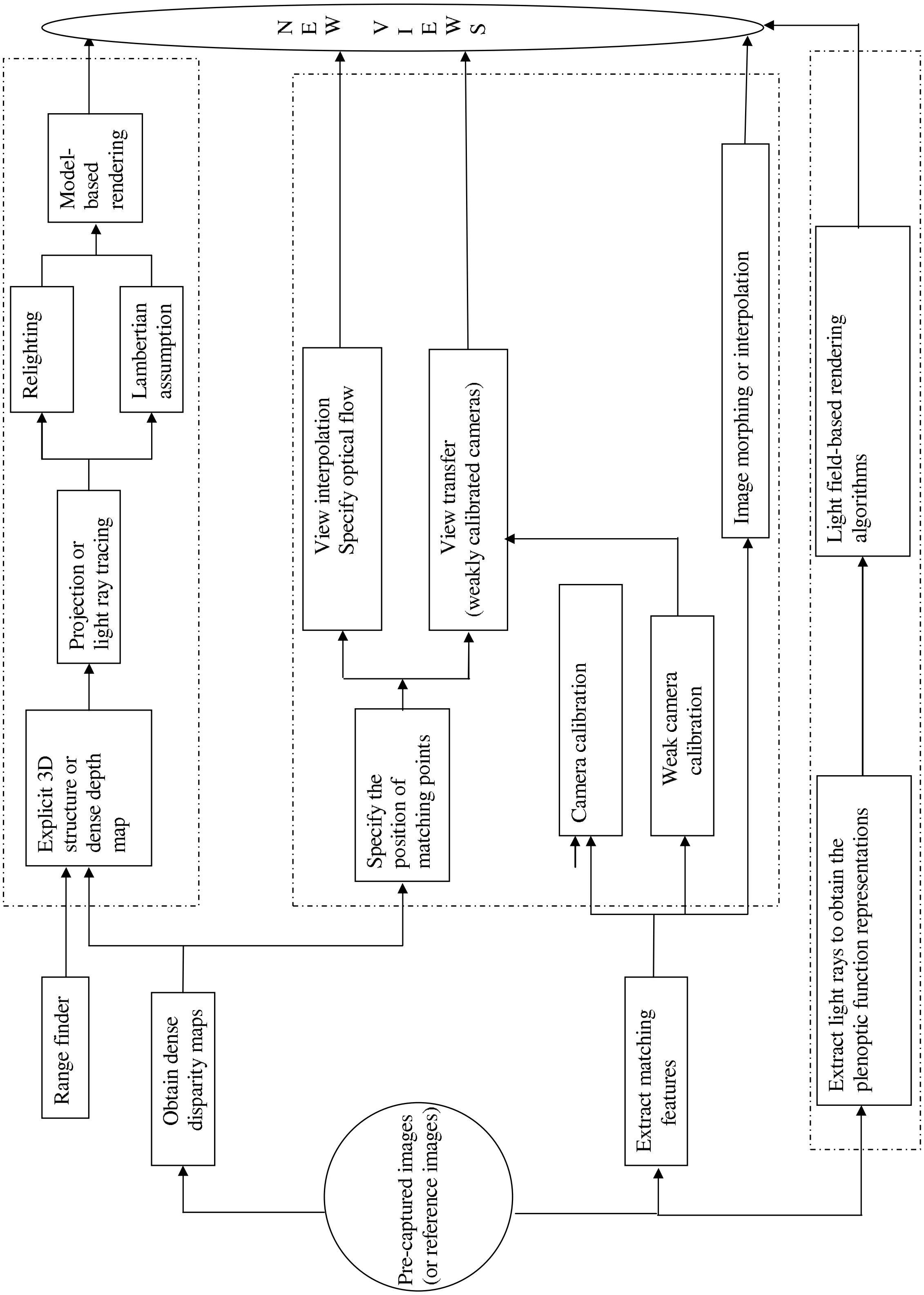

NEW VIEWS
Model-based rendering
Relighting
Lambertian assumption
Projection or light ray tracing
Explicit 3D structure or dense depth map
Range finder
Obtain dense disparity maps
Pre-captured images (or reference images)
View interpolation Specify optical flow
View transfer (weakly calibrated cameras)
Specify the position of matching points
Camera calibration
Weak camera calibration
Image morphing or interpolation
Extract matching features
Light field-based rendering algorithms
Extract light rays to obtain the plenoptic function representations

Figure 1.4: The overview of representations based on implicit plenoptic functions

been proposed [23]. The local geometry of the scene is usually in the form of a depth distribution. The depth values are related to the camera positions in the viewer's coordinate systems. Obviously, this local geometry, or depth distribution, is only helpful for a limited range of viewing positions and directions due to the complex occlusion relationships between different objects in the scene.

The depth values for the correspondent scene points are associated with each pixel in the pre-captured images. A 3D warping algorithm based on dense depth maps has been proposed [24]. The dense depth maps are also used in [25] for texture mapping. A new depth-based representation for IBR namely layered depth image (LDI) was proposed in [26]. In this method, multiple depth values from different viewpoints for a same scene point are associated with one pixel in a certain view for a concise representation in order to deal with the occlusion problem. The representation is extended to a more general one using LDI trees [27].

In addition, the concept of view-dependent geometry [28] and thus the method of view-dependent texture mapping [8] has been proposed with an efficient implementation [29]. The partially reconstructed 3D model [30] is used for view synthesis to improve the rendering quality.

1.1.2 General-view-interpolation approach

One idea for IBR is to jointly apply the techniques of panoramic views [2] and view interpolation [1]. Novel views with similar positions and directions to those of pre-captured images can be interpolated. A more complete framework for this approach is plenoptic modelling [3]. Novel views are interpolated directly from cylindrical panoramas in this method. As a special case, the panoramic video can be used [31] if the navigation is constrained on certain pre-defined paths. Some initial research work using similar constraints appears in [2], [32], [33].

For the general view interpolation approach, the 3D structure or depth values may not be explicitly recovered. The novel views are directly generated from the pre-captured views. In this approach, both the dense disparity maps between pre-captured views, and the camera's relative positions where the pre-captured images are taken, are required. Thus, the scene geometry is still involved in an implicit way. The key requirement for this class of approaches is to determine the relationships of the correspondent matching points between the reference views and then specify the positions of these correspondent points in the novel views. The way to specify the positions of dense matching points in the novel views can be very simple [1] if the camera's relative positions are simple, such as the case of parallel camera views. More generally, the positions of the dense matchings in the novel views are specified through epipolar geometry constraints, using quantities such as the fundamental matrix [14] or the trifocal tensor [15], [16]. This kind of approach is termed view transfer. The epipolar geometry relationships can be computed through a set of reliable matching features, such as corners [34].

View interpolation can also be carried out using sparse matching features. In view morphing [13], the new views are transformed from *one* reference view. Although it is the only possibility when just one reference view (pre-captured image) is available, the most significant contribution in this method is to convert the two-dimensional processing to a one-dimension one after view rectification, so that the view morphing is carried out line-by-line in the horizontal direction. The approach can be easily extended to view interpolation by using the texture from two reference images [35] [36]. The methods can also be easily extended to the methods using dense disparity maps if they are available.

1.1.3 Approach based on light field description

Recently, a new class of approaches that explicitly implement the plenoptic functions for IBR applications has been introduced [17], [4]. The camera's movements when taking pre-captured images are usually specially constrained in order to obtain the sufficient and necessary light rays of the plenoptic functions in a systematic way. The most significant advantage of this approach is that rendering of the novel views can be independent of the scene geometry if a sufficient number of pre-captured images are available. Thus, these methods are classified into the category of rendering without geometric information in [11]. However, geometric information about the scene can still be helpful for interpolation of new light rays in the rendering procedure if it is available, especially when the plenoptic functions are not sufficiently sampled. Due to the difficulties of acquiring plenoptic functions with the full number of dimensions, the plenoptic functions are usually represented with reduced dimensions in the current techniques to simplify the technical requirements. As a consequence, the range of navigation areas and/or the viewing directions may be limited. The approach is considered as restraining the viewing space [12].

It is impossible to pre-capture all light rays in the plenoptic function and it is in many cases not necessary to do so. This is the basic idea behind this approach. Based on the assumption that the light rays do not change along their locus of propagation, the light field modelling techniques aim at using only necessary light rays to represent all the light rays in the plenoptic function. As shown in Fig. 1.5, one light ray passes through P_1, P_2, P_3, P_4, P_5, and P_6. Thus, instead of using six light rays, one light ray is enough to represent all six light rays toward these six positions in the specified direction and in fact all positions along its propagation trace. Thus the techniques of light field modelling use a set of pre-captured images to extract the representative light rays. The rendering of any arbitrary view is the procedure of recombining the properly selected light rays for a specific location and view direction.

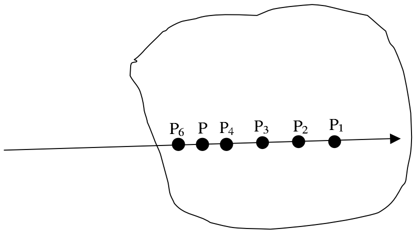


Figure 1.5: A light ray in free space

Essentially, the approach technically provides a new method to obtain the light rays, the basic elements participating in the rendering, which are extracted from the pre-captured images taken in a specific constrained fashion. The approach is thus very convenient for rendering due to the efficient IBR data representations. The conversion from representations based on 3D reconstruction to representations based on light field description are possible. The representations based on light field description can also possibly be obtained by using the pre-captured images together with some other required views which can be interpolated from the pre-captured images.

The concrete techniques of different light-field-based representations are classified according to the dimensions of the plenoptic functions that are actually implemented. More details on different plenoptic function representations with different dimensions can be found in [11] and [12]. The key problem in the techniques based on the light-field description is how to record the representatives of all possible light rays by means of pre-captured images and thus efficiently index each light ray. The major known techniques include panoramas [2], Concentric Mosaics [17], and Light Field Rendering [4].

Panoramic views [2] are the simplest method for the IBR application. Although the panoramas can be captured by a camera with large field of view [37], with a single specially designed optical system, the lens distortions are usually very large. Thus a more attractive approach is to stitch multiple images, giving the so-called image mosaics. These multiple images could be obtained from one moving camera or special multiple camera systems such as the Ladybug camera [38], etc.

If multiple overlapped images are taken by a camera rotated around its projection center or by multiple cameras sharing the same projection center, they usually are warped onto a common virtual imaging surface to minimize the differences in the overlapped area. The commonly used surfaces are cylindrical, spherical, or cubic surfaces. Then the adjacent images with some overlapping area can be stitched [39]. It is very difficult to rotate a camera exactly around its projection center, or to constrain the projection centers of multiple cameras to be at exactly the same position in practice, and thus discontinuities may appear in the stitched images. In order to reduce these discontinuities, research has been carried out to recover the camera's positions and thus to adjust the camera position relationships [40], [41], [42]. Work has also been done to eliminate ghosts due to possible moving objects in the scene [43].

Another kind of image mosaic involves stitching images with multiple projection centers, or manifold mosaic [44]. The key technique here is also try to minimize the discontinuities in the stitched view. A slit camera model with a pipe surface as the common virtual imaging surface was introduced in this work.

The user has only very limited freedom when navigating in a panorama-based virtual environment. With a large number of pre-captured images, Light Field Rendering [4] provides a way to strictly obtain explicit plenoptic functions and allows the user to navigate in a bounded 3D space. In this method, a two-plane reference system is used to index the light rays in the pre-captured images and a 4D plenoptic function is mechanically implemented by precisely controlling the camera's movement. Similar methods but using two spherical surfaces [45], or one spherical surface and one plane [46] as the reference system to index light rays have also been proposed. In addition, an algorithm to speed up the light field rendering was studied in [47]. A technique similar to Light Field Rendering, known as the Lumigraph technique, uses a set of irregularly spaced pre-captured images and the rough scene geometry, obtained in a pre-processing step [5]. The rendering algorithms for the Lumigraph technique are described in [30].

In order to reduce the technical requirement to implement the IBR techniques based on light-field-description, the concentric mosaics technique was proposed in [17]. It is a clever way to reduce the dimensions of the plenoptic functions for scene representations by introducing column-based view synthesis. The columns in the pre-captured images are the basic units participating in the view synthesis. The column-based view interpolation is also used in [48].

1.1.4 Comparison of different approaches and application considerations

In the approaches based on the light-field-description, the plenoptic function is represented by the representative light ray samples which are extracted from the pre-captured images. The light rays, the basic elements that will be used in the rendering, are organized in the viewer's coordinate system. Thus, the rendering is straightforward and relatively easy. The number of pre-captured images is usually large with a great deal of data redundancy. With advanced techniques in data compression, this will not be a significant problem. The key issues for this approach are the technical requirements for implementation. Ideally, the camera positions where the pre-captured images are taken should be precisely located at the pre-determined positions through the use of mechanical control systems. The camera positions may also be estimated. Both the position-control precision and the estimation precision will directly affect the quality of the rendered views.

For the methods using general view interpolation, the geometry of the 3D scene is usually used implicitly and fewer pre-captured images are required. The number of pre-captured images is determined by the ranges of viewing positions and directions that are specified by application requirements. This property makes this approach very flexible. In both view interpolation and view transfer approaches, the intensities and the colors of the same scene points are assumed to be unchanged (the Lambertian assumption) in the related pre-captured images, and that is the basic principle used to obtain the dense disparity maps and the matching features. The approach relies on robust algorithms to obtain precise dense disparity maps, which is a fundamental problem in computer vision. The quality of the rendered novel views is usually significantly affected by the precision of the correspondent matching relationships in the different views.

The methods based on 3D reconstruction models are very efficient and many advanced techniques for model-based rendering in computer graphics can be directly inherited once the 3D models are available. However, the precise and robust 3D model reconstruction is a very difficult problem in which research has been carried out for more than twenty years and much still remains to be done.

On the other hand, the application specifications are the essential considerations before a specific IBR technique is selected. There are three main considerations: the required quality of the rendered views, the constraints on the navigation (in both position and viewing direction), and the cost of a particular technique. The cost includes the technical requirement to obtain the pre-captured images, the quality of data relating to the pre-captured images to be processed and stored, the complexity of the rendering algorithm, etc.

The idea can be illustrated using the panoramas as an example, which are currently very popular in practical applications. The methods to obtain the pre-captured images are very simple and the number of required pre-captured images is low. The rendering algorithms are almost standard for each different panoramic representation (cylindrical, spherical, cubic) with acceptable quality of the rendered novel views. The rendering algorithms are simple and thus can be implemented in real-time. The significant drawback is the limited navigation area. As a consequence, the tradeoff between the above considerations has to be balanced in developing a successful IBR method.

1.2 Thesis Orientation

Considering that it is very difficult and expensive to recover the 3D model and the depth distributions of a scene, the approaches based on reconstruction of 3D scene geometry will not be studied in this thesis. The current algorithms to recover 3D scene geometry are usually not robust and reliable, although the author does believe that the depth information will be helpful in various IBR techniques.

Panoramas are the most basic and important technique for various IBR applications. Panoramic views can be generated from pictures taken by an ordinary camera mounted on a tripod and rotated around its projection center. These multiple images with overlap area between adjacent ones have to be stitched to generate panoramic views. In the view mosaicking technique, the overlap area registration is required before two adjacent views can be stitched. The current research is focused on recovering the camera pose with which the pre-captured images have been taken. The camera-pose recovery is usually based on the different general camera motion models. Complex camera motion models are usually required in such situations, but a more complex model does not necessarily yield better results. A simple, but non-linear model, which is focused on concrete algorithms for panoramic view generation will be studied in this thesis.

View interpolation and view transfer are the basic techniques needed to recover the plenoptic functions using a small number of pre-captured images. Most of the previous approaches are based on dense disparity maps and in the scenario that the novel views are generated from two reference views, in which the correspondent matching relationships are established. In IBR applications, robust algorithms to interpolate novel views from multiple reference views are useful. Thus, we will study view interpolation and view transfer from multiple nearby views with similar imaging directions.

Light Field Rendering and Concentric Mosaics are two special representative techniques using the light field description. By comparing the technical requirements for practical implementation, we find that the Concentric Mosaics (COM) technique is easier to implement. A Simplified Concentric Mosaics (SCOM) technique using non-uniformly distributed pre-captured images is proposed in this thesis, which further reduces the technical requirements for implementation. In the proposed SCOM, the camera positions are estimated from the pre-captured images. In this way, the camera positions where the pre-captured images are taken do not have to be precisely controlled as conventional COM requires. Moreover, the data structure of the COM and SCOM will be compared to illustrate the similarities between the rendering procedures of COM and SCOM.

1.3 Organization of the thesis

In Chapter 2, the mathematical model for IBR, i.e., the plenoptic function, will be presented. The topics studied in this thesis will be illustrated with respect to this model.

In Chapter 3, a new method for registration and stitching of adjacent views in an image mosaic, specifically for cylindrical panoramas, is described. The possible registration errors for the overlap area of two pictures captured by a camera mounted on a tripod at different rotation angles have been analyzed based on a general camera rotation model. Then a novel algorithm is proposed based on both affine adjustment and focal-length adjustments on the optimal strip block where the stitching will be implemented. Matching features in the overlap area have been used as control points to implement the proposed method. The registration is carried out based on the positions of the selected matching features instead of the texture in the overlap area. The stitching errors can be greatly reduced because a narrow strip block is selected instead of the whole overlap area. In addition, a novel algorithm is developed to reduce the accumulated registration errors in the overlap area between the first image and the last image when stitching a set of images that cover 360° view one by one in order to generate cylindrical panoramas.

In Chapter 4, view interpolation from adjacent images is studied. First, a method for view morphing and interpolation based on triangulation is presented. View morphing is regarded here as a basic tool for view interpolation. In the proposed method, the view change is specified through the motions of feature points, which serve as control points. The triangulation of the images, combined with an affine transformation model, has been applied for the texture mapping. The method to interpolate views from two source images has been extended to that from three source images for IBR applications. Then, a novel dense-disparity-based view interpolation algorithm for IBR application is given. The algorithm aims at the scenarios where the traditional dense-disparity-based view interpolation fails to provide good interpolated views. The reasons why the traditional dense-disparity-based view interpolation does not work well are illustrated and the improved algorithm follows.

In Chapter 5, an IBR rendering technique to simplify the implementation of the conventional COM technique, named Simplified Concentric Mosaics (SCOM), has been proposed. In SCOM, the camera positions are estimated from the pre-captured images instead of being precisely controlled. An algorithm based on the stereo technique will be proposed for the camera rotation angle estimation in this special scenario. The algorithm includes several techniques, such as the closed-loop constraint, the ratio fitting method, using total least squares method to solve linear equations, etc. In addition, a pre-processing step to eliminate or reduce the possible vertical offsets and other distortions in the pre-captured images is also proposed, since in a column-based view synthesis technique like the proposed method and the ordinary Concentric Mosaics technique, these vertical offsets and distortions in the pre-captured images will lower the quality of the synthesized images. Thus these methods can be applied on both the proposed SCOM technique and the conventional COM technique.

In addition, the pre-captured image data structures of both COM and SCOM have been illustrated and the comparison has been made. It can be shown that the proposed technique has a similar data structure and thus a similar rendering algorithm as the ordinary COM technique. As a result, it meets our objective that an ordinary user can obtain the COM technique-based image data and plug it into a common COM rendering framework.

The thesis will conclude with a summary of contributions that have been achieved in this thesis research and directions for future work.

Chapter 2

Theoretical framework and background

The goal of IBR is to generate views of an environment from arbitrary locations within a certain navigation area. These views can be regarded as pictures taken by a moving virtual camera at the specified locations. They can be synthesized from the plenoptic function, which is a complete set of light rays of the scene:

$$\mathbf{\Psi} = \{P(\theta, \phi, V_x, V_y, V_z)\} \tag{2.1}$$

where, $P(\theta, \phi, V_x, V_y, V_z)$ denotes the trichomatic color components of one light ray passing through point (V_x, V_y, V_z) in direction (θ, ϕ).

Light rays at a sufficient number of positions and directions must be obtained in order to adequately reconstruct the plenoptic function. These light rays are usually extracted from a set of pre-captured images of the environment. Thus a sufficient number of pre-captured images is needed, and the camera positions and orientation where these pre-captured images are taken must be known.

Thus we see that the plenoptic function relates the pre-captured images to synthesized views. In other words, the novel views are essentially generated from the pre-captured images through the plenoptic function representation, which provides an efficient way to organize the light rays in space.

For a particular IBR application, the requirements are to generate arbitrary novel views: (1) at any position (V_{x0}, V_{y0}, V_{z0}) in a certain 3D space; (2) in any viewing direction (θ_0, ϕ_0) within a certain range. The view direction ranges at different viewing positions are not necessarily the same. In order to simplify the illustration, a fixed viewing range is used here. Assume that the novel views are generated by a virtual camera with field of view F'. An arbitrary view $I_k^{'}, k = 1, 2, 3, ..., K^{'}$ is generated by $P(\theta, \phi, V_{xk}, V_{yk}, V_{zk})$ where $K^{'}$ is the total number of arbitrary views that are required to be generated. (V_{xk}, V_{yk}, V_{zk}) is the projection center of camera capturing $I_k^{'}$ and $(\theta, \phi) \in F'$. The values of the plenoptic function that are defined on the set

$$\mathcal{N}_k = \{(\theta, \phi, V_{xk}, V_{yk}, V_{zk})|(\theta, \phi) \in F'\} \tag{2.2}$$

are required to generate $I_k^{'}$. The ***navigation space*** is defined as $\mathcal{N} = \bigcup_{k=1}^{K^{'}} \mathcal{N}_k$. In another words, any arbitrary view can be generated from values of the plenoptic function on the set $\mathcal{N}$.

On the other hand, the ***capture space*** can be defined in a similar fashion. A physical camera captures a set of image $I_k, k = 1, 2, 3, ..., K$. In the same way, each image I_k gives information about $P(\theta, \phi, V_{xk}, V_{yk}, V_{zk})$ where (V_{xk}, V_{yk}, V_{zk}) is the projection center of camera capturing I_k and $(\theta, \phi) \in F$. F is the camera's field of view. The capture space is defined as $\mathcal{C} = \bigcup_{k=1}^{K} \mathcal{C}_k$ where

$$\mathcal{C}_k = \{(\theta, \phi, V_{xk}, V_{yk}, V_{zk})|(\theta, \phi) \in F\}. \tag{2.3}$$

K is the total number of pre-captured images.

Due to the property of a light ray, i.e., its intensity changes very slowly along its propagation path, there is much redundancy in navigation space $\mathcal{N}$. The values of the plenoptic function at certain different $(\theta, \phi, V_{xk}, V_{yk}, V_{zk})$ can be the same or almost the same. Some IBR techniques such as Light Field Rendering aim to provide more concise representations of the plenoptic function though reducing such redundancy. Similarly, we can infer, by interpolation and extrapolation, values of the plenoptic function at points outside of the capture space $\mathcal{C}$.

Given a particular IBR technique, the positions and shooting directions of the pre-captured images are well planned. Thus, the values of the plenoptic function $P(\theta, \phi, V_{xk}, V_{yk}, V_{zk})$ in navigation space $\mathcal{N}$ can be estimated by given $P(\theta, \phi, V_{xk}, V_{yk}, V_{zk})$ for $(\theta, \phi, V_{xk}, V_{yk}, V_{zk}) \in \mathcal{C}$. $\mathcal{N}$ depends on $\mathcal{C}$. The objective of IBR techniques is to find effective methods on how to select $\mathcal{C}$ in order to maintain the number of the pre-captured image K as small as possible to generate the desired $\mathcal{N}$. Thus, a large number of novel views $K^{'}$ (even infinite in theory) can be synthesized. The capture space $\mathcal{C}$ that can generate $\mathcal{N}$ with minimal number of the pre-captured images is an optimal solution, which is very difficult to obtain in practice.

The light rays that can be extracted from the pre-captured images are discretely distributed in space whereas the plenoptic function is a continuous function of its independent variables. Thus a *discrete capturing space* is essentially obtained to represent the correspondent *capturing space* in order to generate a specified *navigation space.*

Usually, the pre-captured images and the synthesized views are conventional discrete planar images. The relationship between the conventional discrete planar image and its plenoptic function representation is studied first in section 2.1. They are related through the basic element of the plenoptic function, i.e., the light ray.

The panorama is a fundamental and important representation for IBR applications. It can provide some simple functions for some IBR applications and can also serve as the basic representation format for some other IBR applications. Thus, the relationship between a panorama and its plenoptic function representation will be given in section 2.2. View interpolation from the adjacent views will be one of the main topics in this thesis. An extended range of navigation space is obtained by exploring the capturing space, which is achieved through matching corresponding points in different views and then specifying the motions of corresponding 3D points. This scenario will be modelled in the plenoptic function framework in section 2.3. In the last section, the techniques based on light field description will be briefly discussed, providing another way to explore the capturing space.

2.1 Plenoptic function representation of conventional discrete planar images

Conventional discrete planar images can be associated with the values of the plenoptic function at certain positions and in certain viewing directions, and the novel views which are generated from the plenoptic function are also conventional discrete planar images. It is fundamental to illustrate how such conventional discrete planar images are related to the basic elements in the plenoptic function, namely the light rays.

Let W define the area where a planar image $I(x, y)$ is defined and set $I(x, y) = 0, (x, y) \in \mathbb{R}^2 \setminus W$. $I(x, y)$ is generated from the set of light rays toward one specified position $\boldsymbol{V_0}(V_{0x}, V_{0y}, V_{0z})$ within a certain viewing range. The values of plenoptic function within such viewing range can be represented by,

$$\boldsymbol{\Psi_0} = \{P(\theta, \phi, V_x, V_y, V_z) | \theta \in \Theta_0, \phi \in \Phi_0, V_x = V_{0x}, V_y = V_{0y}, V_z = V_{0z}\} \tag{2.4}$$

where sets Θ_0 and Φ_0 determine the ranges of viewing directions which are constrained by the camera's field of view in both horizontal and vertical directions.

Assume there is a one-to-one mapping between (x, y) and (θ, ϕ). We use $\mathcal{Q}$ to denote this mapping relationship, i.e., $\mathcal{Q}:\ (x, y) \mapsto (\theta, \phi)$ or $\mathcal{Q}^{-1}:\ (\theta, \phi) \mapsto (x, y)$. Thus the range of viewing directions is given by $(\Theta_0, \Phi_0) = \mathcal{Q}W = \{(\theta, \phi) | (\theta, \phi) = \mathcal{Q}(x, y), (x, y) \in W\}$.

The relationship between the plenoptic function, i.e., the set of light rays, and the image can be illustrated with Fig. 2.1 by an ideal pinhole camera [49]. In Fig. 2.1, the imaging intensity $I(\lambda, x, y)$ at point A, (x, y), is proportional to the spectral irradiance of the light ray reaching this point, or $P(\lambda, \mathcal{Q}(x, y), V_{0x}, V_{0y}, V_{0z})$. This relationship can be represented as,

$$I(\lambda, x, y) \propto P(\lambda, \mathcal{Q}(x, y), V_{0x}, V_{0y}, V_{0z}),\ (x, y) \in W. \tag{2.5}$$

The conventional discrete image $\hat{I}(m, n)$ (represented in a certain color space) is generated by,

$$\hat{I}_i(m,n) = \int\int\int I(\lambda, m{+}s_1, n{+}s_2)\overline{q}_i(\lambda)a(s_1,s_2)d\lambda ds_1 ds_2,\ (m,n) \in W{\cap}\Lambda,\ i = 1,2,3. \tag{2.6}$$

Here, $I(\lambda, m+s_1, n+s_2)$ should be $I(\lambda, mX+s_1, nY+s_2)$. X and Y are sampling spacings along x and y directions, respectively. For simplicity, we set $X = Y = 1$. The $\overline{q}_i(\lambda)$ are ideally the color matching functions. In practice, they are the sensitivities of three color sensors, such as $r(\lambda)$, $g(\lambda)$ and $b(\lambda)$. $\hat{I}_i(m,n), i = 1,2,3$ represent the three color components of image $\hat{I}(m,n)$, and $a(-s_1,-s_2)$ defines the impulse response function of the camera, or point-spread function (PSF). For an ordinary camera with circular aperture (diameter d) and the focal length f, the PSF is defined as,

$$a(s_1,s_2) = 2\frac{J_1(\pi\frac{\sqrt{s_1^2+s_2^2}}{r_0})}{\pi\frac{\sqrt{s_1^2+s_2^2}}{r_0}} \tag{2.7}$$

where $J_1(\cdot)$ is the first-order Bessel function of the first kind. The parameter r_0 is called the Abbe distance and is determined by the optical property of the camera system. The integration is applied within each sensor element, which usually is a rectangular area. Usually, the sampling density on the lattice Λ, or the density of the sensor elements, should match with the Abbe distance of the optical system; otherwise an extra low-pass anti-aliasing filter has to be introduced before sampling. In practice, the PSF can be modelled as some other functions such as Gaussian function and the relationship between imaging model and observation model has recently been discussed for image magnification application [50].

It should be noted that the capturing space represented by this set of light rays can be much larger than the limited area on the imaging plane because the light ray will not change rapidly along its propagation path.

Thus, the relationship between the conventional discrete image $\hat{I}(m,n)$ and the light rays $P(\theta, \phi, V_x, V_y, V_z)$ can be represented by,

$$\hat{I}_i(m,n) = \int\int\int P(\lambda, \mathcal{Q}(m,n), V_{0x}, V_{0y}, V_{0z}) q_i(\lambda) a(s_1, s_2) d\lambda ds_1 ds_2, \qquad (m,n) \in W \cap \Lambda,\ i = 1,2,3. \tag{2.8}$$

after proper normalization. In the future, we may use image I to represent its three color components of $I_i, i = 1, 2, 3$. The subscript of I will not represent the color component of the image unless it is specified explicitly.

From the above relationship between the conventional planar image and the plenoptic function, we can conclude that a discrete plenoptic function, or plenoptic function defined on a discrete capturing space, is essentially obtained from the discrete pre-captured images. The sampled positions in the capturing space are not uniformly distributed, and the distributions on a certain area are significantly affected by the distance from the camera position to that area. If multiple images are taken, the positions and orientations of the camera where the pre-captured images are taken determine the sampling of the capturing space and the sampled positions can be irregular.

This non-uniform or even irregular sampling of the plenoptic function will affect the quality of the synthesized views, which are generated assuming that the plenoptic function is continuous, and thus the sampling density issue of the capturing space has to be considered when designing the pre-captured acquisition part of an IBR system. However, if the camera positions where the pre-captured images are taken are close to the positions where the novel views will be generated, the problem can be relieved when assuming the physical and virtual cameras have similar parameters (the diameter of the aperture and the focal length). Thus, we will not specifically address this issue in this thesis.

The light ray interpolations have to be carried out when generating novel views from discrete plenoptic functions, or essentially from the pre-captured images. The pre-captured images have undergone lowpass filtering due to the limited camera aperture. Thus the novel views can be obtained from the continuous version I, which is interpolated from $\hat{I}$ using a linear interpolation operator $\mathcal{H}$, i.e. $I(x,y) = (\mathcal{H}\hat{I})(x,y)$, through any suitable interpolation method such as bilinear, bicubic, or spline.

2.2 Panoramic Views

Panoramic views are also formed by groups of light rays, specifically groups of light rays with extended ranges of view directions (θ and ϕ) at one particular location $\boldsymbol{V_0}(V_{0x}, V_{0y}, V_{0z})$. The spherical, cylindrical and cubic panoramas have different viewing direction ranges. For cylindrical panoramas, Θ_0 is defined to be $[0, 2\pi]$ and Φ_0 is determined by the camera's vertical field of view. Theoretically, Θ_0 can be defined from 0 to 2π and Φ_0 can range from $-\pi/2$ to $\pi/2$ for spherical panoramas, which might not be necessary for practical applications. Cubic panoramas provide another method for panoramic representation, which can have the same viewing direction ranges as cylindrical or spherical panoramas. In the following discussion, we will use the cylindrical panorama as an example to illustrate the relationship between panorama and plenoptic function representation, which is similar to the relationship between conventional planar image and plenoptic function representation.

Assume $I_p(x,y)$ is a cylindrical panorama ((x,y) defined in the coordinate system located on the cylindrical surface, as shown in Fig. 2.2). The relationship between the panorama and the light ray is similar to that for planar images,

$$I_p(x,y) = P(\mathcal{Q}_p(x,y), V_{0x}, V_{0y}, V_{0z}),\ (x,y) \in W_p. \tag{2.9}$$

W_p is the area where $I_p(x,y)$ is defined. $\mathcal{Q}_p$ is the mapping relationship between (x,y) and (θ,ϕ). The discrete version of the panorama, denoted by $\hat{I}_p(m,n)$, is actually used in practice after spatially filtering by the optical system and the sensor element of the camera (which could be a virtual camera). This is similar to what we have illustrated in the previous section for the planar conventional image. Thus, the corresponding set of discrete light rays is defined on discrete values of $(\theta,\phi) = \mathcal{Q}_p^{-1}(m,n)$ through $\hat{I}_p(m,n)$. As opposed to the situation with the conventional planar image, the discrete values (θ,ϕ) are uniformly distributed in Θ_0 and Φ_0, respectively. A panoramic view

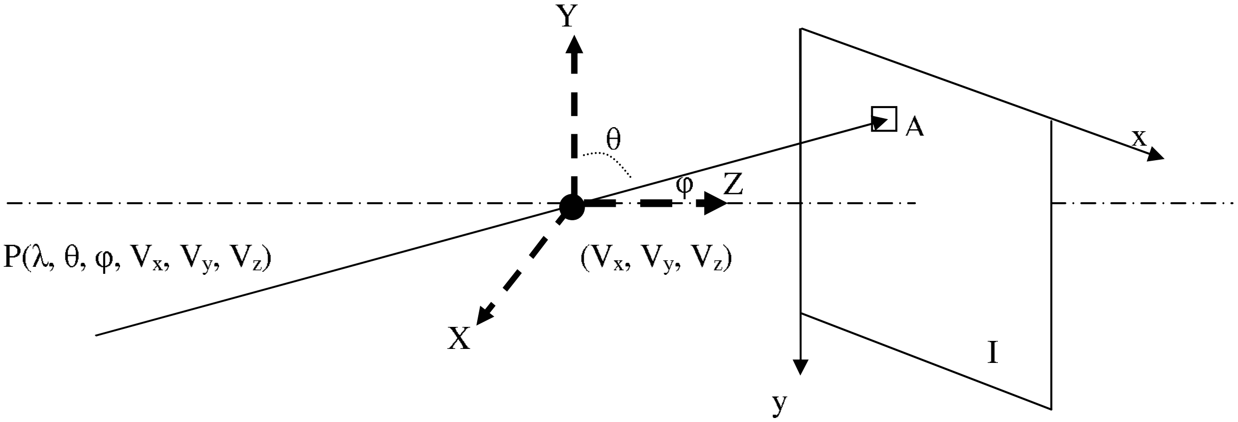


Figure 2.1: basic imaging principle: relationship between plenoptic function and image

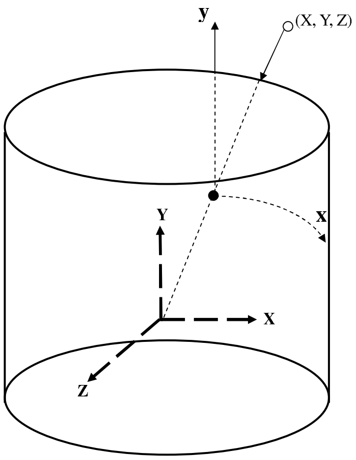


Figure 2.2: The coordinate relationship for warping an image onto a cylindrical surface

represents a restricted plenoptic function and provides some basic IBR functions for navigation. It can generate arbitrary views at one particular position in the navigation space and synthesize approximate views when moving forward and backward along radial directions. These properties can be obtained by analyzing the capturing space that a panorama represents.

The panoramas can be captured by a panoramic camera or generated by a set of images that are taken by an ordinary camera mounted on a tripod and rotated around its imaging center. In this way, the set of light rays represented by a panorama is constructed by some subset of light rays with overlapped definition ranges,

$$\boldsymbol{\Psi_0, p} = \boldsymbol{\Psi_{0,1}} \cup \boldsymbol{\Psi_{0,2}} \cup \boldsymbol{\Psi_{0,3}} ... \cup \boldsymbol{\Psi_{0,i}} ... \cup \boldsymbol{\Psi_{0,N}} \tag{2.10}$$

where,

$$\begin{aligned} \boldsymbol{\Psi_{0,i}} = \{\boldsymbol{P_i}(\theta_i, \phi_i, V_x, V_y, V_z) | \theta_i \in \Theta_i, \phi_i \in \Phi_i, V_x = V_{0x}, V_y = V_{0y}, V_z = V_{0z}\}, \\ i = 1, 2, ..., N. \end{aligned} \tag{2.11}$$

For cylindrical panoramas, the horizontal viewing ranges $\Theta_i, i = 1, 2, ..., N$ are overlapped. For spherical panoramas, both horizontal viewing ranges $\Theta_i, i = 1, 2, ..., N$ and vertical viewing ranges $\Phi_i, i = 1, 2, ..., N$ are overlapped.

Each of the $\boldsymbol{\Psi_{0,i}}$ can be obtained from the conventional planar images $I_i(x, y)$ $(i = 1, 2, ..., N)$ captured by an ordinary camera. However, the (θ, ϕ) values have to be uniformly distributed in the ranges Θ_0 and Φ_0 for panoramic views. Thus, the conventional planar images $I_i(x, y)$ have to be processed so that all θ_i and ϕ_i, determined by the processed images, are uniformly distributed in their correspondent ranges. These processing procedures are termed warping, which map the conventional images onto a common surface such as a cubic, cylindrical or spherical surface, as if they were captured by a camera with its imaging sensor on this common surface. The warping process can be denoted by $\mathcal{W}$ and $(\mathcal{W}I_i)(x, y)$ is obtained after applying warping on $I_i(x, y)$. The warping processing here is essentially a transformation between different coordinate systems. For example, it is a transformation from an ordinary coordinate system on a plane to a coordinate system defined on a cylindrical surface when generating cylindrical panoramas in our example. Further details will be discussed in following chapter.

The warped images can then be stitched together to generate the panoramic image $I_p(x, y)$. Due to the warping process, there are some distortions that can be observed in panoramic views. After a panorama at a certain position has been obtained, any view at the given position within a certain viewing-direction range can be obtained by selecting the correspondent light rays for a particular viewing range. However, the current synthesized views are in a warped state, or defined in the coordinate system on a cylindrical surface. Thus, a de-warping process $\mathcal{W}^{-1}$, $I(x, y) = \mathcal{W}^{-1} I_p(x, y), (x, y) \in W_1$ has to be applied on particular viewing ranges of (θ, ϕ) to generate the planar images without warping distortions. The area W_1 is defined by the viewing ranges (θ, ϕ).

The overall procedure for using panoramas for IBR applications can be briefly described in the following. The overlapping conventional discrete images $\hat{I}_i(m, n)$, $(m, n) \in W_i \cap \Lambda$ are taken by a camera mounted and rotated on a tripod around the camera center or by a multi-camera system sharing the same imaging center. The warped images $(\mathcal{W}\mathcal{H}\hat{I}_i)(x, y)$ can be obtained from $(\mathcal{H}\hat{I}_i)(x, y)$. In practice, the discrete images $(\mathcal{W}\mathcal{H}\hat{I}_i)(m, n)$ (obtained from $(\mathcal{W}\mathcal{H}\hat{I}_i)(x, y)$ using the method similar to equation (2.6)) are used for stitching to generate the discrete panorama $I_p(m, n)$, $(m, n) \in W_p \cap \Lambda$. A novel view for a particular viewing range (Θ_k, Φ_k), $((\Theta_k, \Phi_k) \subset (\Theta_0, \Phi_0))$, can be generated by de-warping $(\mathcal{H}\hat{I}_p)(x, y)$, $(x, y) \in W_{p,k}$ in the particular range $W_{p,k} = \mathcal{Q}(\Theta_k, \Phi_k)$, i.e. $(\mathcal{W}^{-1}\mathcal{H}\hat{I}_p)(x, y)$. The discrete image $\hat{I}(m, n), (m, n) \in W_k \cap \Lambda$ is actually obtained. W_k (the range where $\hat{I}$ is defined) is determined by $W_{p,k}$ (the correspondent range that $\hat{I}_p$ is selected). When the virtual camera moves along the radial direction from point V_{0x}, V_{0y}, V_{0z}, the correspondent views can be approximately obtained through zooming. In the zooming mode, the light rays in the plenoptic function representation on the area where it is not defined are approximated by the values within its capturing space.

The panoramic views provide restricted plenoptic functions for some basic IBR applications which have been widely used recently. All novel views at one specified position, and views when moving forward and backward in the radial direction from the specified position, can be synthesized from a panorama. The panoramic views also serve as a basic technique for other IBR approaches due to their efficient way to represent all light rays toward one particular position. For these reasons, panoramas are considered as an important and basic technique in IBR and will be discussed further in Chapter 3.

2.3 View interpolation from adjacent views (partial plenoptic-function representations)

With the panoramic view, arbitrary views in all possible viewing directions, i.e., all possible values of (θ,ϕ), at one specified position $\boldsymbol{V_0}$ can be obtained through the plenoptic function represented by a panorama.

Since it is impossible to capture all images at every point, i.e. all possible values of $\boldsymbol{V}(V_x, V_y, V_z)$ within the navigation area, a straightforward method is to capture some representative images to generate the panoramas at some particular locations within the navigation area and synthesize views (or panoramas) at other specified locations through view interpolation. In this way, methods to determine the camera positions (V_{0x}, V_{0y}, V_{0z}) where the representative images are taken and algorithms for view interpolation are required.

The idea of view interpolation is based on the Lambertian assumption. For a given scene point S, its imaging point in image I_1 is located at $(x_{1,s}, y_{1,s})$ and at $(x_{2,s}, y_{2,s})$ in image I_2. Images I_1 and I_2 have similar viewing directions and can be conventional planar images or parts of panoramas. If image I_1 is captured at $\boldsymbol{V}_{0,1}(V_{0x,1}, V_{0y,1}, V_{0z,1})$ and Image I_2 is captured at $\boldsymbol{V}_{0,2}(V_{0x,2}, V_{0y,2}, V_{0z,2})$, then the correspondent light rays represented by these two images are $P(\theta_1, \phi_1, V_{0x,1}, V_{0y,1}, V_{0z,1})$

and $P(\theta_2, \phi_2, V_{0x,2}, V_{0y,2}, V_{0z,2})$, respectively. The Lambertian assumption can be expressed as,

$$P(\mathcal{Q}(x_{1,s}, y_{1,s}), V_{0x,1}, V_{0y,1}, V_{0z,1}) \approx P(\mathcal{Q}(x_{2,s}, y_{2,s}), V_{0x,2}, V_{0y,2}, V_{0z,2}). \tag{2.12}$$

If the depth of any scene point (the distance from the scene point to $\boldsymbol{V}_{0,1}$) which can be projected to image I_1 is known, the view I_2 can be generated from image I_1 based on the above Lambertian assumption. This is the technique termed view morphing [13], which has been widely used in computer graphics. In the above condition, position $(x_{2,s}, y_{2,s})$ can be calculated from the $\boldsymbol{V}_{0,1}$, $\boldsymbol{V}_{0,2}$, $(x_{1,s}, y_{1,s})$, the distance from S to $\boldsymbol{V}_{0,1}$, and the virtual camera's internal parameters (the image center and the focal length) based on simple geometric relationships. Similar relationships for imaging-position transfer can be applied for any other points in image I_2. Thus, the transformation from discrete image $\hat{I}_1(m,n)$ to $\hat{I}_2(m,n)$ is the change of sampling structure.

$$\hat{I}_2(m,n) = (\mathcal{H}\hat{I}_1)(m + x_{2,1}(m,n), n + y_{2,1}(m,n)), (m,n) \in W \cap \Lambda \tag{2.13}$$

where W is the range where I_2 is defined. Point $(m + x_{2,1}(m,n), n + y_{2,1}(m,n))$ in I_1 is the imaging position of a scene point whose imaging position in I_2 is (m,n). This assumes that this scene point is visible from both image I_1 and I_2. This is not always the case if occlusions happen during the transition from one view to the other. Point $(m + x_{2,1}(m,n), n + y_{2,1}(m,n))$ in I_1 and point (m,n) in I_2 are a pair of correspondences. The array of all $(x_{2,1}(m,n), y_{2,1}(m,n))$ forms the *disparity* map. However, the depth distribution (or geometry) of the scene is very difficult to obtain in practice. Thus, view interpolation and view transfer are usually used. In the following context, we assume both I_1 and I_2, and the disparity map between I_1 and I_2 are known.

Traditionally, view interpolation has been studied based on two source images [1]. The views at the positions between the two camera positions where two source images are taken can be interpolated. Assume that a viewing position $\boldsymbol{V}_i = (V_{0x,i}, V_{0y,i}, V_{0z,i})$ is located on the line $\boldsymbol{V}_{0,1}\boldsymbol{V}_{0,2}$ (between $\boldsymbol{V}_{0,1}$ and $\boldsymbol{V}_{0,2}$). The view I_i at this position has a similar viewing direction to I_1 and I_2 if I_1 and I_2 are conventional planar images. Further assume that the imaging point of the scene point S in image I_i is located at $(x_{i,s}, y_{i,s})$. $(x_{i,s}, y_{i,s})$ is usually determined by a correspondence motion model (i.e. uniform translation assumption) through $(x_{1,s}, y_{1,s})$, $(x_{2,s}, y_{2,s})$ and the position of $\boldsymbol{V}_i$ on line $\boldsymbol{V}_{0,1}\boldsymbol{V}_{0,2}$, or determined by the techniques in computer vision such as using the fundamental matrix [14]. Based on the above Lambertian assumption,

$$\begin{aligned}P(\mathcal{Q}(x_{i,s}, y_{i,s}), V_{0x,i}, V_{0y,i}, V_{0z,i}) = w_1(i, x_{1,s}, y_{1,s}) \cdot P(\mathcal{Q}(x_{1,s}, y_{1,s}), V_{0x,1}, V_{0y,1}, V_{0z,1}) + \\ w_2(i, x_{2,s}, y_{2,s}) \cdot P(\mathcal{Q}(x_{2,s}, y_{2,s}), V_{0x,2}, V_{0y,2}, V_{0z,2}) \quad (2.14)\end{aligned}$$

$w_1(i, x_{1,s}, y_{1,s})$ and $w_2(i, x_{2,s}, y_{2,s})$ denote different weights applied on a particular pair of light rays from different views, and $w_1(i, x_{1,s}, y_{1,s}) + w_2(i, x_{2,s}, y_{2,s}) = 1$ for a particular i. In a similar way, all light rays associated with image I_i can be obtained. Thus the interpolated view $\hat{I}_i$, at position $\boldsymbol{V}_i = (V_{0x,i}, V_{0y,i}, V_{0z,i})$, can be synthesized by

$$\begin{aligned}\hat{I}_i(m, n) = w_1(i, m + x_{i,1}(m, n), n + y_{i,1}(m, n)) \cdot (\mathcal{H}\hat{I}_1)(m + x_{i,1}(m, n), \\ n + y_{i,1}(m, n)) + w_2(i, m + x_{i,2}(m, n), n + y_{i,2}(m, n)) \\ \cdot (\mathcal{H}\hat{I}_2)(m + x_{i,2}(m, n), n + y_{i,2}(m, n)), (m, n) \in W \cap \Lambda \quad (2.15)\end{aligned}$$

Here, W is the range where image I_i is defined. $\hat{I}_1$ and $\hat{I}_2$ are the discrete versions of image I_1 and I_2 that are used in practice. $(m + x_{i,1}(m, n), n + y_{i,1}(m, n))$ and (m, n) are a pair of correspondences located in image I_1 and I_i, respectively. Similarly, $(m + x_{i,2}(m, n), n + y_{i,2}(m, n))$ and (m, n) are a pair of correspondences located in image I_2 and I_i, respectively. $(x_{i,1}(m, n), y_{i,1}(m, n))$ and $(x_{i,2}(m, n), y_{i,2}(m, n))$ are specified through the disparity map between I_1 and I_2 from a feature motion model or a projection model.

For IBR applications, the navigation areas are usually within a 2D area rather than on a 1D line. One strategy is to divide the 2D navigation area into a set of sub-areas, the combination of which cover the original navigation area. The basic sub-area could be chosen in triangular shape. If three source images (with similar viewing directions) are taken at camera positions which are located at three vertices of a basic triangular area, then the views in the similar viewing directions from any position within this triangular area may be possibly synthesized from these three reference views.

This leads to our research topic on view interpolation from three source images. Obviously, other shapes rather than the triangle can be used in the above approach. In that way, the view interpolation will be carried out on multiple (larger than three) source images for a general case. Then arbitrary views can be synthesized when a virtual camera moves on a 2D plane, given there are three (or more) pre-captured images with viewing directions and camera positions similar to the virtual camera's current viewing direction and position. The situation where view interpolation is carried out using three (or more) pre-captured views can be formulated in a similar way to view interpolation based on two views. The correspondent movement can be determined through the trifocal tensor [15], [16].

The navigation space is extended by exploring the correlations between the light rays in the capturing space during view interpolation. In this way, fewer pre-captured images are required, thus simplifying the methods for acquisition of pre-captured images. As a tradeoff, the dense disparity maps, or motion field, which are usually difficult to obtain precisely, are required. Thus the way to get a precise motion field between pre-captured images will be studied in the thesis.

One way to get the motion field, at least as an initial estimation, is using matching features. Although the methods to obtain precise sparse matching features are not robust for all scene scenarios, they are more reliable than current dense disparity estimation algorithms and much research work has been carried out in this field. The dense disparity maps then can be estimated from the sparse matching features in triangulation-based approaches, where a local smoothness constraint is usually applied for approximation. Thus a novel-view-synthesis algorithm is proposed by means of triangulation through sparse feature matchings. Further details will be presented in Chapter 4.

2.4 Techniques based on light field description: plenoptic function representations

In the above plenoptic function representation, the light rays $P(\theta, \phi, V_x, V_y, V_z)$ are determined by their directions (different values of θ and ϕ) in a coordinate system located at the camera's position (V_x, V_y, V_z). As we mentioned before, the potential navigation space can be much larger than the capture space because the light rays do not change rapidly along their propagation directions. This is the idea behind the methods based on light field description. For example, consider two camera positions at $\boldsymbol{V}_1(V_{x,1}, V_{y,1}, V_{z,1})$ and $\boldsymbol{V}_2(V_{x,2}, V_{y,2}, V_{z,2})$. The light rays in $\boldsymbol{V}_1\boldsymbol{V}_2$ and $\boldsymbol{V}_2\boldsymbol{V}_1$ directions are defined twice in the above plenoptic function representation. The situations are applied between any two camera positions in 3D space. Thus more efficient ways to represent the light rays may exist and are of interest for IBR applications.

In the Light Field Rendering technique, the 4D plenoptic function is represented with the assistance of two parallel planes E_1 and E_2 as shown in Fig. 2.3. Any arbitrary light ray $l(k)$ (as long as it is not parallel with E_1 and E_2) can be determined by two points $M(x_1(k), y_1(k))$ (in the E_1 plane) and $N(x_2(k), y_2(k))$ (in the E_2 plane), or in the form of $\tilde{P}(x_1(k), y_1(k), x_2(k), y_2(k))$. $\tilde{P}(x_1, y_1, x_2, y_2)$ is the plenoptic function represented by the Light Field Rendering technique. In other words, each light ray can be specified by its intersections with E_1 and E_2. Assume that a camera takes images when moving on one plane such as the E_1 plane and the E_2 plane is the focal plane (or imaging plane) of the camera. In image I_k taken at position N (in Fig. 2.3), every pixel corresponds to one light ray passing through N and all light rays with the same $(x_2(k), y_2(k))$ coordinates are recorded in this image. I_k is correspondent to the light ray set $\{\tilde{P}(x_1, y_1, x_2, y_2) | x_2 = x_2(k), y_2 = y_2(k), (x_1, y_1) \in W\}$, where W is determined by the camera's field of view in both horizontal and vertical directions.

We use 2-parallel-planes as an example to illustrate the idea of the 4D plenoptic function representation. If only two parallel planes are used, the light rays that can be represented have to intersect with these two parallel planes. In practice, more than one set of 2-parallel-planes are used to cover the light rays in all directions [4], [5], [46]. In a more general situation, the camera may move on any other surface as long as the camera positions along that surface can be obtained precisely [45]. Then each light ray can be determined in a general 2-surface parametrization framework [30].

The camera has to be precisely controlled during pre-captured-images acquisition in the techniques using the above 2-surface parametrization approach. This is very expensive for an ordinary user to obtain the pre-captured images, and potentially prevents the related techniques from being widely used.

If the navigation activities are constrained within a certain plane, then the above 2-surface parametrization can be simplified. The idea can be illustrated in Fig. 2.4. The line L_1 is the camera movement trajectory when the pre-captured images are taken. Plane E_2 is the camera's imaging plane. In Fig. 2.4, an arbitrary light ray $l(k)$ comes from the correspondent scene point S. This light ray is not captured. The light ray $l^{'}(k)$ comes from the same scene point S but it intersects with both plane E_2 (at $N^{'}$) and $L1$ (at M). Thus, light ray $l^{'}(k)$ is captured. Due to the fact that both light rays come from the same scene point, the values of plenoptic function for these two light rays are similar. By approximation, we can use one (e.g. $l^{'}(k)$) to replace the other (e.g. $l(k)$).

In order to find light ray $l^{'}(k)$ (which is used to replace light ray $l(k)$) in the plenoptic function database, we need to know the position $N^{'}$. It is a simple geometric relationship if the 3D depth distribution is known. However, the reconstruction of 3D depth distribution of the scene is not a trivial task. Thus, further approximation is required. Usually, a constant depth of the scene, or the infinite depth of the scene, can be assumed to simplify the rendering procedure. This approximation will cause vertical distortion in the synthesized views. However, this approach can significantly simplify the procedure to obtain the required light rays in the pre-captured image acquisition, since the camera movement is only required to be precisely controlled in

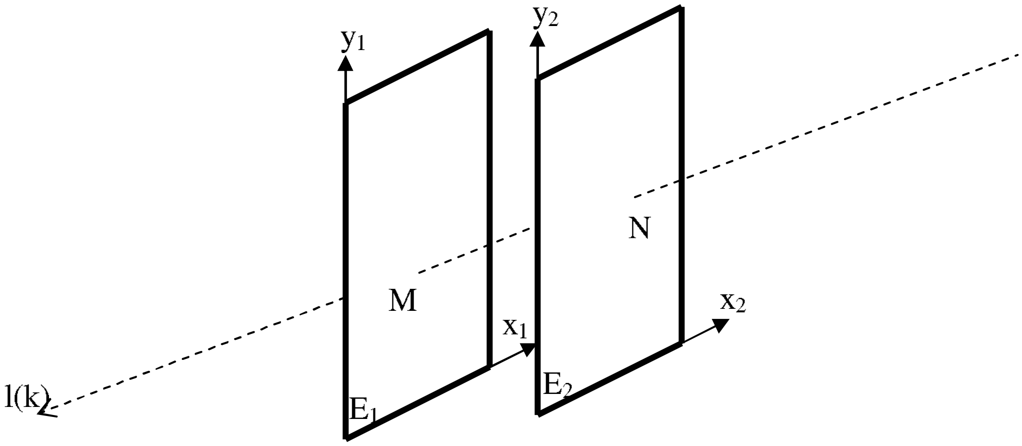


Figure 2.3: 2-plane parametrization of light rays for Light Field Rendering

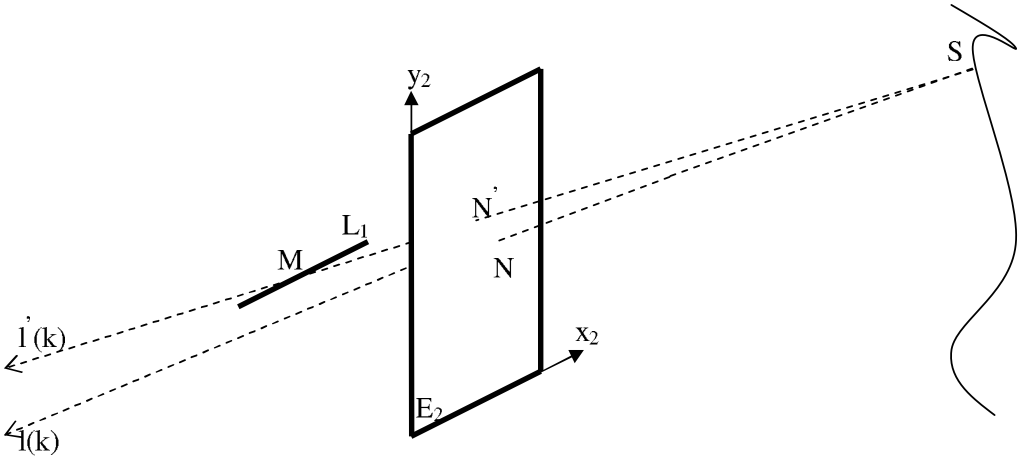


Figure 2.4: Illustration of 3D parametrization of light rays

one dimension. Moreover, experiments show that the distortion in the synthesized view is acceptable. This may be because the average depth of the scene is larger than the depth variation in the usual case. Further details on this topic can be found in [51].

This is the idea behind the methods in [47], [52] [53] and Concentric Mosaics [17], which uses a column-based view interpolation strategy to synthesize the novel views. In Concentric Mosaics (COM), the camera is precisely controlled to move on a circle in the acquisition of the pre-captured images.

2.5 feature points and matching features between multi-views

One major task in IBR techniques is to find the relationships between views taken at different camera positions and in different imaging directions. These views could be pre-captured images or the synthesized views (as if taken by a virtual camera). One common way to extract the relationship between multi-views is to identify the corresponding points in different views. The corresponding points (sometimes referred to as correspondences [34]) are the imaging points of the same scene point in different images that are taken at different positions and/or in different directions. After we identify enough correspondences, the relationships between these multiple views can be retrieved and thus the relationship between the camera positions and imaging direction where these multiple views were taken can be estimated. There are basically two ways to represent corresponding points in different views. One is disparity maps and the other is matching features. Matching features are extensively used in several of the proposed techniques in this thesis. We briefly introduce the concept of matching features in the following.

It is possible to find reliable correspondences using some "special" points in the images. This leads to the concept of *feature points*. Features are some specific patterns, or points with "certain properties" from the image processing point of view.

The procedure to identify the features (or corners) in an image is *feature detection* (or *corner detection*). After detecting the feature points in the individual images, the corresponding points can be identified by matching the feature points in different views. This is the procedure of *feature matching.* The resulting corresponding features are called *matching features.* Thus, matching features are special and reliable corresponding points between multiple views. This is one major reason that they are widely used in computer vision and image processing.

By definition, feature points must have some special properties to distinguish them from other points. The selection criteria for features include [54]: localization, robustness, sensitivity, stability and complexity. These criteria are used to evaluate different feature detection algorithms. Usually, a computable metric for measurement is proposed in the feature detection algorithms to represent the mentioned special properties (texture patterns) around the point in the image. The most straightforward way is to measure the intensity variations within a local window around a point [55]. However, this measurement is not very reliable. The derivatives of image intensity are more reliable measurement to detect features. Some measurements based on the the derivatives of image intensity can be found in [56], [57], [58]. Among these different algorithms for corner detection, Harris corner [59] detection is one of the most successful methods and thus has been widely used. In the Harris corner detection algorithm, a corner response function is defined as

$$R = D(M) - \alpha \cdot T(M)^2 \tag{2.16}$$

where $M = [\begin{smallmatrix} I_x^2 & I_xI_y \\ I_yI_x & I_y^2 \end{smallmatrix}]$ is the autocorrelation matrix. I_x and I_x represent the first order derivatives of image intensity along x and y direction, respectively. α is a constant. $D(\cdot)$ denotes the calculation of the determinant of a matrix. $T(\cdot)$ denotes the calculation of the trace of a matrix. A similar approach can be found in [60].

Recent work has been extended to detect more difficult scenarios, i.e., to detect scale and affine invariant features. For example, the multi-scale representation [61] is used to detect scale invariant features (SIF) [62]. The research work on affine invariant feature detection includes the algorithms based on the second moment matrix [63], the methods to extract affine invariant regions in the images [64], etc.

In order to match the corresponding features in different views, a unique identifer is required to distinguish the feature points from one with each other. The mathematical representations of such identifers are named *feature descriptors*. They are used to match features in different views. In various matching algorithms, the feature descriptors are first computed. The measurement metric, such as cross-correlation [65], is used to identify corresponding features for comparisons of these feature descriptors in different views. Different algorithms to represent and compare feature descriptors are proposed in the literature, which may use shape or/and edge information of the image [66], histogram statistics (such as SIFT descriptor [67], [62]) of a local area in the image, etc.

In this thesis, we will mainly use Harris corner and cross-correlation matching algorithm. This combination approach has relatively low computing complexity and it is sufficient to generate reliable matching features for the imaging conditions that we deal with.

Chapter 3

Generation of panoramic views using image stitching

Pictures with a large field of view (FOV), such as panoramic views, are a simple but important scene representation format in image-based rendering (IBR) [11]. Common approaches to obtain such pictures are to stitch several pictures with overlapping area, taken by a camera moving in a specific fashion [2], [39], [68], [42], [69], [70], [32], or by a panoramic camera with multiple optical systems and imaging sensors, such as the Ladybug panoramic camera [38], or other types of special optical systems such as omnidirectional cameras [71], fisheye lens [37], etc. More general scenarios for view mosaics can be found in [44], [72], [73], [74], [75].

Overlap-area registration [76] is the most important step that must be accomplished before stitching. In this chapter, we will propose a new efficient algorithm for panoramic view generation.

3.1 Overview of previous work

Traditional image registration methods only consider the translations along horizontal and vertical directions. When depth variations are small compared with the average depth, such as in most outdoor-scene applications, the registration of adjacent images is not a big problem and most of the available commercial software, (e.g., [77], [102]), can perform well. However, when the depth variation is large compared to the average depth, as in most indoor-environment applications, this depth variation will cause significant problems for overlap area registration between the adjacent views. In order to reduce the overlap-area registration errors, research work on overlap-area registration has been reported in the literature to recover the camera motion based on different models. Under this framework, various methods to register the overlap area of two adjacent images have been proposed based on an 8-parameter perspective transformation [40], a polynomial transformation with more freedom [41], and other geometric corrections [42]. A comparison of some commonly-used parametric models for camera motion recovery can be found in [78].

In the previous work, the registration and the stitching are usually implemented based on the whole overlap area, which can be problematic because the overlap area may be too large for one global transformation to yield a good result. Thus local corrections have to be used for ghosting cancelation [40], which makes the registration more complex and introduces the possibility of causing further discontinuities among local blocks and even distortions, especially when dealing with high resolution pictures. In addition, 360° cylindrical panoramas have become very common recently. In this situation, views from different directions are captured by a camera that is mounted on a tripod and rotated around its optical center. The adjacent images, with some overlap area, are then stitched together. However, the accumulated registration errors between the first image and the last image when generating a 360° panorama are usually very large. This issue was reported in [40], although it may not be a serious issue when only translations are used for registration. In [40], these accumulated registration errors in the vertical direction are modeled as a global rotation. The global rotation angle is then evenly assigned to the transformations on each of the pre-captured images after finding this global rotation angle through an initial tentative registration. Then registrations and stitching are carried out again considering the assigned rotation angles in each transformation. This will affect the registration results and the way to assign the accumulated global rotation angle on each transformation is tentative. Several trials may be required to experimentally approach a good result. Thus, no reliable work has been found in the literature to reduce the potential accumulated errors when generating a 360° panorama. In conclusion, the problem of effective overlap-area registration still has many open problems.

In this chapter, a new registration model has been proposed which jointly applies affine adjustment and focal-length adjustment. A matching-feature-based overlap area registration method has been used for implementation. A novel algorithm based on a coordinate-system-transformation model has been proposed to reduce accumulated errors when generating 360° panoramic views. In addition, the idea of stitching on an optimal strip area has been proposed. A multi-resolution stitching algorithm is also proposed to avoid potential large intensity changes when stitching two adjacent images on a narrow strip area.

3.2 Camera movement model and basic techniques

In order to stitch two adjacent images, the difference between the two views in the overlap area must be minimized to obtain stitched images with good quality. From a general camera rotation model [42], we find that camera pose and movement deviating from the ideal case is the main cause of registration error [79]. Affine adjustment and focal-length adjustment are used in the proposed algorithm to reduce the registration error in the overlap area caused by such non-ideal camera pose and movement. The affine transformation on a discrete image is defined first. Then, cylindrical warping is discussed in order to introduce the nonlinear part of the proposed model. In addition, the precise definition of the overlap area is given, followed by a brief discussion of feature detection and matching in the overlap area.

3.2.1 A general camera rotation model for analysis of registration errors

When a camera is mounted on a tripod and captures images at different rotation angles, its motion usually deviates from a pure rotation about the camera projection center. We refer to the camera's projection center as the camera position in the following sections. A general camera rotation model [42] is illustrated in Fig. 3.1. It is not easy to physically locate the projection center of a real camera. Thus the camera center usually moves along a circle (the circle may not even be on the level plane), as shown by the dashed arc in Fig. 3.1, rather than remaining fixed at a given point, such as O in Fig. 3.1. Assume that two images I_1 and I_2 are captured from adjacent viewpoints (O_1 and O_2 in Fig. 3.1) with some overlap area. Using similar notation to [80], we assume that image I_1 is captured with a camera having a certain frame of reference F_1 with origin at the camera center O_1 and I_2 is captured with a camera having frame of reference F_2 with origin O_2. The coordinate vector of a scene point

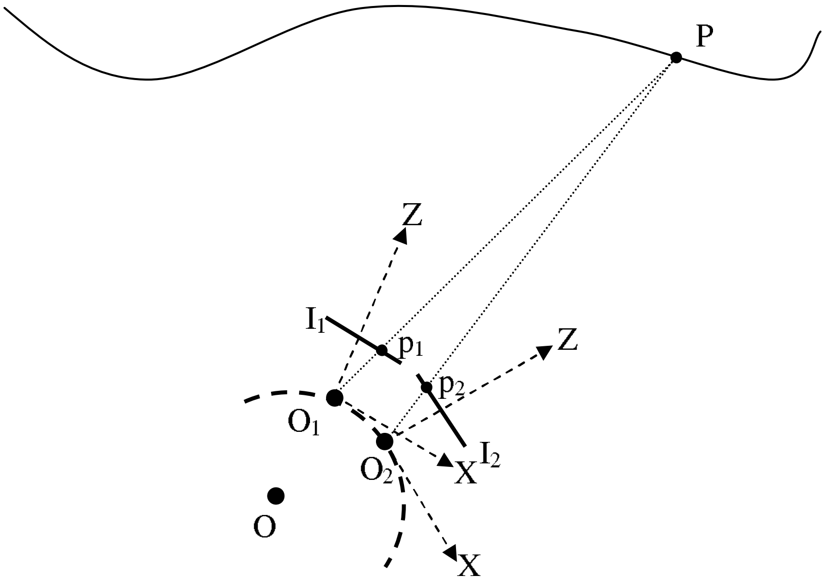


Figure 3.1: The general camera rotation model

P in frame F_i $(i = 1, 2)$ is denoted ${}^{F_i}P = [{}^{F_i}P_x {}^{F_i}P_y {}^{F_i}P_z]^T$. Then ${}^{F_2}P = {}^{F_2}_{F_1}R \cdot {}^{F_1}P + {}^{F_2}O_1$, where ${}^{F_2}_{F_1}R$ is a rotation matrix and ${}^{F_2}O_1$ is the coordinate vector of O_1 in frame F_2 [80]. Using the pinhole camera model, the point P is projected to points p_1 and p_2 in F_1 and F_2 respectively, given by $p_i = f_i \cdot {}^{F_i}P / {}^{F_i}P_z$, where f_1 and f_2 are the focal lengths associated with the two camera positions. Combining these equations, we obtain

$$w_{12} \begin{bmatrix} p_{2x} \\ p_{2y} \\ f_2 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} + {}^{F_2}O_{1x} / {}^{F_1}P_z \\ r_{21} & r_{22} & r_{23} + {}^{F_2}O_{1y} / {}^{F_1}P_z \\ r_{31} & r_{32} & r_{33} + {}^{F_2}O_{1z} / {}^{F_1}P_z \end{bmatrix} \begin{bmatrix} p_{1x} \\ p_{1y} \\ f_1 \end{bmatrix} \tag{3.1}$$

where $w_{12} = f_1 \cdot {}^{F_2}P_z / (f_2 \cdot {}^{F_1}P_z)$, ${}^{F_2}O_1 = [{}^{F_2}O_{1x} \quad {}^{F_2}O_{1y} \quad {}^{F_2}O_{1z}]^T$ and $p_i = [p_{ix} \quad p_{iy} \quad p_{iz}]^T$ $(i = 1, 2)$. r_{ij} $(i, j = 1, 2, 3)$ are the elements of 3 by 3 matrix ${}^{F_2}_{F_1}R$. The key observation here is that the transformation to describe the positions of corresponding points in the overlap area of the two adjacent images is depth-dependent. Perfect registration can only be achieved with perfect 3D reconstruction of the depth distribution of the environment, which is difficult to implement in practice.

3.2.2 Discrete affine transformation

Affine transformations are normally defined on continuous-space images. For a discrete-space image I, the affine transformation is defined as follows. Assume that a continuous-space image corresponding to I can be obtained using a linear interpolation operator $\mathcal{H}$. Then the discrete affine transformation operator with parameter vector $\boldsymbol{t}$, denoted $\mathcal{A}_{\boldsymbol{t}}$, is defined by $(\mathcal{A}_{\boldsymbol{t}} I)(\boldsymbol{x}) = (\mathcal{H} I)(\boldsymbol{T}(\boldsymbol{x} - \boldsymbol{x}_0) + \boldsymbol{d} + \boldsymbol{x}_0), \boldsymbol{x} \in \Lambda$, where $\boldsymbol{T} = \left[\begin{smallmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{smallmatrix}\right]$ and $\boldsymbol{d} = [d_1 \ d_2]^T$. We define the parameter vector $\boldsymbol{t} = [t_{11} \ t_{12} \ t_{21} \ t_{22} \ d_1 \ d_2]^T$ and $\boldsymbol{x}_0$ is an arbitrary reference point. The image center $\boldsymbol{x}_c = (x_c, y_c)$ will be selected as the reference point, so we will not explicitly specify $\boldsymbol{x}_0$ in the following sections unless necessary. Λ is the sampling lattice where the image $(\mathcal{A}_{\boldsymbol{t}} I)(\boldsymbol{x})$ is defined. Of course, $(\mathcal{H} I)(\boldsymbol{x}^{'})$ is only computed at the points $\boldsymbol{x}^{'} = \boldsymbol{T}(\boldsymbol{x} - \boldsymbol{x}_c) + \boldsymbol{d} + \boldsymbol{x}_c, \boldsymbol{x} \in \Lambda$, using any suitable interpolation method such as bilinear, bicubic, or spline. The affine transformation of coordinates on $\mathbb{R}^2$ can also be represented as an operator $\mathcal{S}_{\boldsymbol{t}}$ where

$\mathcal{S}_{\boldsymbol{t}}\boldsymbol{x} = \boldsymbol{T}(\boldsymbol{x} - \boldsymbol{x}_c) + \boldsymbol{d} + \boldsymbol{x}_c$. Thus, we have $(\mathcal{A}_{\boldsymbol{t}}I)(\boldsymbol{x}) = (\mathcal{H}I)(\mathcal{S}_{\boldsymbol{t}}\boldsymbol{x})$. In practical situations, $\boldsymbol{T}$ is non-singular, so that $\mathcal{S}_{\boldsymbol{t}}$ is invertible. The inverse is also an affine transformation, with $\mathcal{S}_{\boldsymbol{t}}^{-1}\boldsymbol{x}^{'} = \boldsymbol{T}^{-1}(\boldsymbol{x}^{'} - \boldsymbol{x}_c) - \boldsymbol{T}^{-1}\boldsymbol{d} + \boldsymbol{x}_c$.

3.2.3 Warping the image onto a cylindrical surface

In order to reduce the difference between two adjacent images in the overlap area, the images should be mapped onto a common surface such as a cylindrical or spherical surface, as if they were captured by a camera with its imaging sensor on this common surface. We choose here a cylindrical surface as the common imaging surface since it is a popular choice to generate 360° panoramas. The warped images are referred to as *cylindrically-warped* images. The cylindrical radius is usually chosen to be the focal length of the planar camera. This choice makes the geometric relationships between the source images and the warped images straightforward and thus the warping algorithm is easy to implement. Using similar notation as above, the cylindrical warping is denoted as $\mathcal{W}_f$ with parameter f, which is the focal length of the camera (assuming that the camera is rotated exactly around its optical center and the focal length is unchanged for adjacent images). Then, the warped image is $(\mathcal{W}_f I)(\boldsymbol{x}) = (\mathcal{H}I)(\boldsymbol{x}^{'}), \boldsymbol{x} \in \Gamma$, where Γ is a lattice in the cylindrical coordinate system. The relationship between $\boldsymbol{x} = [x\ y]^T$ and $\boldsymbol{x}^{'} = [x^{'}\ y^{'}]^T$ can be represented as (obtained from [39]) $x^{'} = f \cdot \tan((x - x_c)/f) + x_c$ and $y^{'} = (y - y_c)/\cos((x - x_c)/f) + y_c$.

This one-to-one nonlinear mapping between $\boldsymbol{x}$ and $\boldsymbol{x}^{'}$ can also be represented by an operator $\mathcal{U}_f$ on $\mathbb{R}^2$, so that $(\mathcal{W}_f I)(\boldsymbol{x}) = (\mathcal{H}I)(\mathcal{U}_f\boldsymbol{x})$ with $\boldsymbol{x}^{'} = \mathcal{U}_f\boldsymbol{x}$. The corresponding inverse mapping defines the de-warping procedure that will be used later. The inverse of the above relationship is given by $x = f \cdot \tan^{-1}((x^{'} - x_c)/f) + x_c$ and $y = f \cdot (y^{'} - y_c)/(\sqrt{(x^{'} - x_c)^2 + f^2}) + y_c$, or $\boldsymbol{x} = \mathcal{U}_f^{-1}\boldsymbol{x}^{'}$. Operator $\mathcal{U}_f^{-1}$ denotes the sampling structure change from the warped image to the de-warped image in the de-warping transformation, $I(\boldsymbol{x}) = (\mathcal{H}(\mathcal{W}_f I))(\mathcal{U}_f^{-1}\boldsymbol{x})$.

One fixed focal length can only be used to warp all images if the camera was rotated exactly around its projection center and the camera focal length was unchanged during the capture procedure, because this focal length is the radius of the cylindrical surface. However, the camera usually does not rotate around its projection center, as illustrated in Fig. 3.1. The optimal virtual common imaging surface for view mosaicking is no longer a cylindrical surface with one fixed radius, or focal length. Thus, the focal length will be assumed variable and can be adjusted for each image. In practice, the real shape of the virtual common imaging surface may be too difficult to recover. Adjustment of focal length alone may be insufficient to recover the practical virtual common imaging surface. In section (3.3.2), we will define the optimal focal length with respect to minimization of the registration errors in the overlap area.

3.2.4 Definition of the overlap area between adjacent images

Assume that I_1 and I_2 are two original adjacent images where I_1 is to the left of I_2. I_1 and I_2 are defined on sampling lattice Λ. Let W_1 define the area where image I_1 is defined and set $I_1(\boldsymbol{x}) = 0, \boldsymbol{x} \in \mathbb{R}^2 \setminus W_1$. Similarly, let W_2 define the area where image I_2 is defined with $I_2(\boldsymbol{x}) = 0, \boldsymbol{x} \in \mathbb{R}^2 \setminus W_2$. In both cases, $\boldsymbol{x}$ is in the image coordinate system, with origin at the top left of the image. The approximate registration of image I_1 and I_2 through a simple shifting technique is given by

$$\hat{\boldsymbol{d}}_0 = \arg\min_{\boldsymbol{d}_0 \in \Lambda} \frac{1}{|W_{12}(\boldsymbol{d}_0) \bigcap \Lambda|} \sum_{\boldsymbol{x} \in W_{12}(\boldsymbol{d}_0) \bigcap \Lambda} |I_1(\boldsymbol{x}) - I_2(\boldsymbol{x} - \boldsymbol{d}_0)|, \tag{3.2}$$

where $W_{12}(\boldsymbol{d}_0) = W_1 \bigcap (W_2 + \boldsymbol{d}_0)$ is the overlap between two regions W_1 and $(W_2 + \boldsymbol{d}_0)$. $|W_{12}(\boldsymbol{d}_0) \bigcap \Lambda|$ represents the number of pixels within area $W_{12}(\boldsymbol{d}_0)$. This is the traditionally-used image registration method [81], [3], [39]. The overlap area between images I_1 and I_2 is defined as $W_{12}(\hat{\boldsymbol{d}}_0) = W_1 \bigcap (W_2 + \hat{\boldsymbol{d}}_0)$. The sub-image of I_1 in the overlap area is $I_{1\text{com}}$, where $I_{1\text{com}}(\boldsymbol{x}) = I_1(\boldsymbol{x}), x \in W_{12,\Lambda}(\hat{\boldsymbol{d}}_0)$ and $W_{12,\Lambda}(\hat{\boldsymbol{d}}_0) = W_{12}(\hat{\boldsymbol{d}}_0) \bigcap \Lambda$. Similarly, $I_{2\text{com}}(\boldsymbol{x}) = I_2(\boldsymbol{x} - \hat{\boldsymbol{d}}_0), x \in W_{12,\Lambda}(\hat{\boldsymbol{d}}_0)$ for the sub-image of I_2 in the overlap area.

The above registration processing is defined on the original images, so that W_1, W_2 and $W_{12}(\hat{\boldsymbol{d}}_0)$ are rectangular in shape. If the affine or/and warping transformations are applied on I_2, then W_2 and $W_{12}(\hat{\boldsymbol{d}}_0)$ may no longer be rectangular. We use W'

to denote the general non-zero area of a transformed image I, so that we denote the overlap area of transformed images by $W'_{12}(\hat{\boldsymbol{d}}'_{\boldsymbol{0}}) = W'_1 \bigcap(W'_2 + \hat{\boldsymbol{d}}'_{\boldsymbol{0}})$.

3.2.5 Matching-feature detection in the overlap area

Matching features are detected in images $I_{1\text{com}}$ and $I_{2\text{com}}$ in order to implement the feature-based approach for registration [82]. We simply match Harris corners [59] based on correlation. The matching relationships between the Harris corners in two sub-images are found through neighbor area searching based on a normalized correlation criterion. The epipolar constraints can be used to guide the matching process [83]. A publicly available software package that includes similar functions can be found on Projection Vision Toolkit (PVT) website [84].

For each feature point, or Harris corner P, located at $\boldsymbol{x}$ in $I_{1\text{com}}$, its matching point in $I_{2\text{com}}$ is searched from among the Harris corners located in a window centered at $\boldsymbol{x}$ in image $I_{2\text{com}}$. This window is the neighborhood searching window. The block-based normalized correlations between every Harris corner within the neighborhood searching window in $I_{2\text{com}}$ and the feature point P in $I_{1\text{com}}$ are calculated. The feature point with maximum normalized correlation is selected and its normalized correlation value is compared with a threshold to determine if it is a good matching point for P. Since the images $I_{1\text{com}}$ and $I_{2\text{com}}$ are very similar to each other, the neighborhood searching window can be set to a relatively small size, which makes matching relatively easy and reliable. More details on the above feature matching algorithm can be found in [85].

The relationship between $I_{1\text{com}}$ and $I_{2\text{com}}$ is represented by two sets of matching feature points $\text{MP}_1 = \{\boldsymbol{x}_{1,n} | n = 1, 2, ..., N\}$ and $\text{MP}_2 = \{\boldsymbol{x}_2(\boldsymbol{x}_{1,n}) | n = 1, 2, ..., N\}$ in $I_{1\text{com}}$ and $I_{2\text{com}}$, respectively. For any feature point $\boldsymbol{x}_{1,n} \in \text{MP}_1$, its matching point in $I_{2\text{com}}$ is denoted as $\boldsymbol{x}_2(\boldsymbol{x}_{1,n}) \in \text{MP}_2$, where N is the total number of matching features in the common region. In the following section, we will use the position of these feature points to determine the transformations of the images I_1 and I_2 for overlap area registration.

Some recently developed more reliable matching features, such as SIFT corners [62], PCA-SIFT [67], etc., can also be used. The algorithms for Harris corner detection are relatively simple and thus save computations. For those overlap areas where there are few or no features that can be detected, the matching features can be selected from dense disparity maps. Usually, stitching of overlap areas having little texture and thus few features, is easy to achieve with few artifacts.

3.3 Overlap area registration using matching features as control points

In this section, we introduce our methods to reduce the registration errors. The affine transformation is used as one technique to adjust the camera pose changes, together with a nonlinear focal-length adjustment procedure. The objective function in the proposed optimization model aims at minimizing the residual registration errors in order to obtain a seamless mosaic view after stitching. Thus the 'optimal' focal lengths obtained do not necessarily have any significant physical meaning.

3.3.1 Camera pose adjustment through affine transformation

The affine adjustment is applied on the *cylindrically-warped* images. For example, assume two *cylindrically warped* images are $\mathcal{W}_{f_0}I_1$ and $\mathcal{W}_{f_0}I_2$. Here, f_0 is the *a priori* estimated focal length of the camera.

In the texture-based approach [79], also developed as part of this thesis research, the affine adjustment on $\mathcal{W}_{f_0}I_2$ is determined by searching for the parameter vector $\hat{\boldsymbol{t}}_2$ that minimizes the average of absolute differences over the overlap area,

$$\hat{\boldsymbol{t}}_2 = \arg\min_{\boldsymbol{t}_2} \frac{1}{|W'_{12}(\hat{\boldsymbol{d}}'_{\mathbf{0}})\bigcap\Gamma|} \sum_{\boldsymbol{x}\in W'_{12}(\hat{\boldsymbol{d}}'_{0})\cap\Gamma} |\mathcal{W}_{f_0}I_1(\boldsymbol{x}) - \mathcal{A}_{\boldsymbol{t}_2}(\mathcal{W}_{f_0}I_2)(\boldsymbol{x})| \tag{3.3}$$

where $W'_{12}(\hat{\boldsymbol{d}}'_{\mathbf{0}})$ is the overlap between the images $\mathcal{W}_{f_0}I_1$ and $\mathcal{W}_{f_0}I_2$ as defined in Section 3.2.4.

For the feature-matching approach, the corresponding optimal affine transformation is given by

$$\hat{\boldsymbol{t}}_2 = \arg\min_{\boldsymbol{t}_2} \frac{1}{N} \sum_{n=1}^{N} |\mathcal{U}_{f_0}^{-1}\boldsymbol{x}_{1,n} - \mathcal{S}_{\boldsymbol{t}_2}^{-1}(\mathcal{U}_{f_0}^{-1}\boldsymbol{x}_2(\boldsymbol{x}_{1,n}))|^2, \tag{3.4}$$

where $\mathcal{U}_{f_0}^{-1}$ and $\mathcal{S}_{\boldsymbol{t}_2}^{-1}$ define the forward mapping for cylindrical warping and affine transformations, respectively.

In order to simplify the illustration of the solution, we use $\boldsymbol{x}_{1,n,\mathcal{U}_{f_0}^{-1}}$ to denote $\mathcal{U}_{f_0}^{-1}\boldsymbol{x}_{1,n}$ and use $\boldsymbol{x}_{2,n,\mathcal{U}_{f_0}^{-1}}$ to denote $\mathcal{U}_{f_0}^{-1}\boldsymbol{x}_2(\boldsymbol{x}_{1,n})$. $\boldsymbol{x}_{2,n,\mathcal{U}_{f_0}^{-1}}$ is a 2-D vector of the coordinate, which can be represented as $\boldsymbol{x}_{2,n,\mathcal{U}_{f_0}^{-1}} = [x^{(1)}_{2,n,\mathcal{U}_{f_0}^{-1}} \; x^{(2)}_{2,n,\mathcal{U}_{f_0}^{-1}}]^T$. If we define a new matrix

$$X_{2,n,\mathcal{U}_{f_0}^{-1}} = \begin{bmatrix} x^{(1)}_{2,n,\mathcal{U}_{f_0}^{-1}} & x^{(2)}_{2,n,\mathcal{U}_{f_0}^{-1}} & 0 & 0 & 1 & 0 \\ 0 & 0 & x^{(1)}_{2,n,\mathcal{U}_{f_0}^{-1}} & x^{(2)}_{2,n,\mathcal{U}_{f_0}^{-1}} & 0 & 1 \end{bmatrix}, \tag{3.5}$$

then we obtain $\mathcal{S}_{\boldsymbol{t}_2}^{-1}(\mathcal{U}_{f_0}^{-1}\boldsymbol{x}_2(\boldsymbol{x}_{1,n})) = X_{2,n,\mathcal{U}_{f_0}^{-1}}\boldsymbol{t}_2$. Thus, equation (3.4) can be reformed as

$$\hat{\boldsymbol{t}}_2 = \arg\min_{\boldsymbol{t}_2} \frac{1}{N} \sum_{n=1}^{N} |\boldsymbol{x}_{1,n,\mathcal{U}_{f_0}^{-1}} - X_{2,n,\mathcal{U}_{f_0}^{-1}}\boldsymbol{t}_2|^2, \tag{3.6}$$

This is a standard least-squares problem with solution [86],

$$\hat{\boldsymbol{t}}_2 = \frac{1}{N}[\sum_{n=1}^{N} X^H_{2,n,\mathcal{U}_{f_0}^{-1}} X_{2,n,\mathcal{U}_{f_0}^{-1}}]^{-1}[\sum_{n=1}^{N} X^H_{2,n,\mathcal{U}_{f_0}^{-1}}\boldsymbol{x}_{1,n,\mathcal{U}_{f_0}^{-1}}] \tag{3.7}$$

More details on the problem formulation can be found in [87]. It can easily be solved using the singular value decomposition (SVD) method or other related algebraic methods (usually $N \gg 3$).

The optimal parameter vector $\hat{\boldsymbol{t}}_2$ is selected to update the current $\mathcal{W}_{f_0}I_2$ by $\mathcal{A}_{\hat{\boldsymbol{t}}_2}(\mathcal{W}_{f_0}I_2)$. The positions of matching features are also updated following the optimal transformation. An affine adjustment can equivalently be applied on image $\mathcal{W}_{f_0}I_1$ to match $\mathcal{W}_{f_0}I_2$.

3.3.2 Focal-length adjustment method

The optimal focal length associated with each image for cylindrical warping is selected as a refinement so as to further minimize the registration error. In the proposed algorithm, the focal-length adjustment is applied on original or de-warped images after some initial geometric transformation (such as the above affine adjustment) has been applied on two adjacent images. For example, assume that we want to apply focal-length adjustment on image I_2, i.e., to find the optimal focal length associated with I_2 that can minimize the overlap registration error with I_1.

First, we warp I_1 with some fixed focal length f_1. It can be the focal length initially estimated from camera calibration, or it can be the focal length obtained from a previous optimization of focal length applied on image I_1. The estimated focal length does not need to be very accurate, which makes the calibration inexpensive. Then, the search for an optimal focal length $\hat{f}_2$ for I_2 using image intensity matching yields [79]

$$\hat{f}_2 = \arg\min_{f_2} \frac{1}{|W'_{12}(\hat{\boldsymbol{d}}'_{\boldsymbol{0}})\bigcap\Gamma|} \sum_{\boldsymbol{x}\in W'_{12}(\hat{\boldsymbol{d}}'{}_0)\cap\Gamma} |\mathcal{W}_{f_1} I_1(\boldsymbol{x}) - \mathcal{A}_{\hat{\boldsymbol{t}}_2}(\mathcal{W}_{f_2} I_2)(\boldsymbol{x})| \tag{3.8}$$

where $W'_{12}(\hat{\boldsymbol{d}}'_{\boldsymbol{0}})$ is the overlap area between the images $\mathcal{W}_{f_1}I_1$ and $\mathcal{W}_{f_{20}}I_2$ and $\mathcal{A}_{\hat{\boldsymbol{t}}_2}$ is the optimal affine transformation. f_{20} is the initial guess of f_2, which also can be the focal length initially estimated from camera calibration, or it can be the focal length from a previous optimization of focal length adjustment applied on image I_2 in an iterative procedure.

The above optimization involves a large amount of computation. The proposed alternative approach is to find the optimal focal length using the positions of matching features in the overlap area. The optimization model for the feature-based approach is then

$$\hat{f}_2 = \arg\min_{f_2} \frac{1}{N} \sum_{n=1}^{N} |\mathcal{U}_{f_1}^{-1}\boldsymbol{x}_{1,n} - \mathcal{S}_{\hat{\boldsymbol{t}}_2}^{-1}(\mathcal{U}_{f_2}^{-1}\boldsymbol{x}_2(\boldsymbol{x}_{1,n}))|^2, \tag{3.9}$$

which is a scalar optimization problem. Because warping transformation $\mathcal{U}_{f_2}^{-1}$ is not a linear transformation, the analytic solution of the above equation could be very complex. Thus, the algorithm of 1D searching around an initial value f_2 is used. The initial value f_2 may be obtained from the manual of camera's optical system or from a simple calibration. If the optimizations are implemented in an iterative way, the initial value of the current iteration is the optimized result from the last iteration.

After the optimal focal length $\hat{f}_2$ is found, warping with this focal length is applied on I_2 for updating the positions of the features in the feature-based approach, for further processing. As we mentioned, focal-length adjustment is only applied on de-warped images in the proposed algorithm.

In the matching-feature approach, areas with more matching features get better registration. Usually, these areas are rich in texture and good stitching is required. On the other hand, we can give different weights to the registration errors for matching features at different locations in equation (3.4) and (3.6) to adjust residual registration errors at different locations in the overlap area. This observation also supports the following idea: a better registration result can be obtained if the matching features on a narrow strip area are chosen (special weights assignment), as will be done later.

3.3.3 The overall overlap area registration algorithm

The overall optimization model to register the overlap area between two adjacent images using the texture-based approach is

$$[\hat{\boldsymbol{t}}_2, \hat{f}_1, \hat{f}_2] = \arg \min_{\boldsymbol{t}_2, f_1, f_2} \frac{1}{|W'_{12}(\hat{\boldsymbol{d}}'_{\boldsymbol{0}}) \bigcap \Gamma|} \sum_{\boldsymbol{x} \in W'_{12}(\hat{\boldsymbol{d}}'_{\boldsymbol{0}}) \cap \Gamma} |\mathcal{W}_{f_1} I_1(\boldsymbol{x}) - \mathcal{A}_{\boldsymbol{t}_2}(\mathcal{W}_{f_2} I_2)(\boldsymbol{x})| \quad (3.10)$$

where $W'_{12}(\hat{\boldsymbol{d}}'_{\boldsymbol{0}})$ is the overlap area between the images $\mathcal{W}_{f_{10}} I_1$ and $\mathcal{W}_{f_{20}} I_2$. f_{10} and f_{20} are initial values of f_1 and f_2, respectively. In the feature-based approach, the corresponding optimization model is

$$[\hat{\boldsymbol{t}}_2, \hat{f}_1, \hat{f}_2] = \arg \min_{\boldsymbol{t}_2, f_1, f_2} \frac{1}{N} \sum_{n=1}^{N} |\mathcal{U}_{f_1}^{-1} \boldsymbol{x}_{1,n} - \mathcal{S}_{\boldsymbol{t}_2}^{-1}(\mathcal{U}_{f_2}^{-1} \boldsymbol{x}_2(\boldsymbol{x}_{1,n}))|^2. \quad (3.11)$$

Finding jointly optimal $\hat{\boldsymbol{t}}_2, \hat{f}_1, \hat{f}_2$ requires the solution of a complex nonlinear optimization problem. Instead, the following alternating optimization algorithm can be used iteratively:

1) set $f_{10} = f_{20} = f_0$ and $g = 0$.

2) The images I_1 and I_2 are warped using focal lengths f_{10} and f_{20}, respectively. The warped version of image I_1 is $\mathcal{W}_{f_{10}} I_1$ and that of I_2 is $\mathcal{W}_{f_{20}} I_2$.

3) Apply affine adjustment on image $\mathcal{W}_{f_{20}} I_2$ to obtain $\mathcal{A}_{\hat{\mathbf{t}}_2}(\mathcal{W}_{f_{20}} I_2)$ after registering the overlap area between $\mathcal{W}_{f_{10}} I_1$ and $\mathcal{W}_{f_{20}} I_2$.

4) Apply focal-length adjustment on I_1 to obtain $\mathcal{W}_{\hat{f}_1} I_1$, and based on which the focal-length adjustment can be applied on I_2 to obtain $\mathcal{W}_{\hat{f}_2} I_2$ after de-warping $\mathcal{A}_{\hat{\mathbf{t}}_2}(\mathcal{W}_{f_{20}} I_2)$.

5) $g = g + 1$; stop? if not, set $f_{10} = \hat{f}_1$ and $f_{20} = \hat{f}_2$, go to 3).

The conditions to terminate the iterations can be: (1) the registration errors are small enough compared to a predefined threshold; or (2) the change of the registration errors is small enough compared to a predefined threshold; or (3) the number of the iterations g has reached a predefined value. In practice, we used g to terminate the iterations. The actual implementations were carried out using the feature-based approaches. The globally optimal parameters for equation (3.8) may not necessarily be found in the above alternating optimization. However, the simulation results show that the solutions obtained are generally good enough for the current applications.

3.4 Choosing a narrow strip block for stitching to minimize the discontinuities

The observation that the final stitching of the two images can be implemented just on a narrow strip area is important. The idea of stitching on a strip block has been appeared in [72] but the approach is based on a large quantities of images (video sequence). We perform the optimization on all possible strip blocks with fixed size in the overlap area, instead of over the whole overlap area, and thus a better registration result can be obtained. The updated overall optimization can be expressed as,

$$[\hat{\boldsymbol{t}}_2, \hat{f}_1, \hat{f}_2, \hat{i}] = \arg \min_{\boldsymbol{t}_2, f_1, f_2, i} \frac{1}{|W'_{12,i}(\hat{\boldsymbol{d}}'_{\mathbf{0}}) \bigcap \Gamma|} \sum_{\boldsymbol{x} \in W'_{12,i}(\hat{\boldsymbol{d}}'_{\mathbf{0}}) \cap \Gamma} |\mathcal{W}_{f_1} I_1(\boldsymbol{x}) - \mathcal{A}_{\boldsymbol{t}_2}(\mathcal{W}_{f_2} I_2)(\boldsymbol{x})| \tag{3.12}$$

where the strip block $W'_{12,i}(\hat{\boldsymbol{d}}'_{\mathbf{0}})$ is a sub-area of the overlap area with fixed size, i.e., $W'_{12,i}(\hat{\boldsymbol{d}}'_{\mathbf{0}}) \subset W'_{12}(\hat{\boldsymbol{d}}'_{\mathbf{0}})$. The subscript i denotes the location where the strip block starts. We are looking for a strip block within the overlap area that gives the minimum registration error after the proposed affine adjustment and focal-length adjustment have been carried out on the image pair I_1 and I_2. Parameter vector $\hat{\boldsymbol{t}}_2$ and parameters $\hat{f}_1$ and $\hat{f}_2$ on this strip block are the optimal ones. In practice, the alternating optimization method can also be used by finding the optimal $\hat{\boldsymbol{t}}_2$, $\hat{f}_1$, and $\hat{f}_2$ for the whole overlap area and then searching for the optimal strip area within the whole overlap area.

In the feature-matching-based implementation, the following observations have to be considered:

- Due to the irregular distribution of the feature points in the overlap area, the strip block for registration should be large enough to contain a sufficient number of feature points.

- The strip block where the registration is carried out is not necessarily the same as the block where the stitching will be implemented; the former can be larger and contain the latter.

- There may be no matching features in the best strip block for stitching.

As a consequence, we define two types of strip blocks: **strip blocks for registration** and **strip blocks for stitching**. The optimal affine transformation parameters and optimal focal length are obtained based on the position registration of the matching features in the strip block for registration and the location of the optimal strip block for registration is recorded. The strip block for registration should be large enough to contain sufficient matching features.

If the overlap area of two adjacent views is not very large, we can simplify the algorithm by choosing one reasonable strip block for registration near the central part of the overlap area. In this case, we will not search for the optimal strip block for registration but simply search for the optimal strip block for stitching. In our implementation, the width of the strip block for registration was usually selected to be five to six times larger than that of the strip block for stitching. This is not necessarily optimal but it gives good results and it reduces the computations. In this way, the optimization on the strip block for registration including both focal-length adjustment and affine transformation is

$$[\hat{\boldsymbol{t}}_2, \hat{f}_1, \hat{f}_2] = \arg \min_{\boldsymbol{t}_2, f_1, f_2} \frac{1}{N_i} \sum_{n \in \xi_i} |\mathcal{U}_{f_1}^{-1} \boldsymbol{x}_{1,n} - \mathcal{S}_{\boldsymbol{t}_2}^{-1}(\mathcal{U}_{f_2}^{-1} \boldsymbol{x}_2(\boldsymbol{x}_{1,n}))|^2 \tag{3.13}$$

where ξ_i is a set which contains an index list of all matching features that are located in the selected strip block for registration. N_i is the number of matching features in ξ_i. If we only select one strip block for registration, then $i = 0$. This is the case we used here, or only one strip block for registration to obtain optimal parameters for focal-length adjustment and affine transformation. Otherwise, the overlap area is divided into many strip blocks for registration in one image (e.g. I_1) if the overlap area is very large. For the i-th strip block for registration, ξ_i is a set which contains an index list of the matching features, where the feature points belonging to this image (I_1) are within this strip (i-th) block. These strip blocks for registration could be overlapped with each other. Then, the optimization can be carried out regarding to strip blocks for registration, or i.

Within the strip area for registration, we search for an optimal narrow strip block for stitching, where the texture registration error is minimized. The strip blocks are of fixed size and can overlap with each other. The location of the optimal strip block for stitching can be obtained from

$$\hat{j} = \arg \min_{j} \frac{1}{|W_{r,j} \bigcap \Gamma|} \sum_{\boldsymbol{x} \in W_{r,j} \bigcap \Gamma} |\mathcal{W}_{\hat{f}_1} I_1(\boldsymbol{x}) - \mathcal{A}_{\hat{\boldsymbol{t}}_2}(\mathcal{W}_{\hat{f}_2} I_2)(\boldsymbol{x})| \tag{3.14}$$

where $W_{r,j} \subset W_r$ is one possible strip block for stitching, with j standing for its start position in horizontal direction. W_r is the strip block for registration. It can be the optimal strip block for registration $W^{'}_{12,\hat{i}}(\hat{\boldsymbol{d}}'_{\mathbf{0}})$, or specified as above, depending on the size of overlap area.

In the algorithm, the position of the finally selected optimal strip block for stitching is determined by the texture difference on each possible strip block instead of the difference between the positions of the matching feature points. The reason is that the visual stitching discontinuities will be visible through texture discontinuities.

It is possible that a sharp intensity change might appear when stitching on a narrow strip area if the luminance change between two adjacent images is very large. For this kind of situation, a multi-resolution-based stitching method is suggested. Each image for stitching is first decomposed into two sub-band components through a complementary pair of 2D filters. The bandwidth of the lowpass filter is very narrow. Then, the lowpass components of two images are stitched on the whole overlap area and the highpass components of two images are stitched on the narrow strip area. After obtaining the lowpass and highpass components of the stitched images, the final stitched image can be obtained by adding its lowpass component and highpass component. In this way, sharp luminance changes can be avoided in the stitched images and the stitching quality is good due to the proposed stitching method on a narrow strip area. The method has been tested with successful results. The stitching is implemented through blending. The correspondent weighted pixels in the stitching area but from different source images are added to generate the transition area in the stitched images. The weights change linearly along the horizontal direction in the stitching areas. The summation of correspondent weights at any pixel in the stitching area equals one. Further details on this blending method can be found in [39]. Other stitching algorithms [88] can also be used.

3.5 Resolution and computation considerations

Each transformation essentially changes the sampling structure where the discrete image is defined. Thus interpolation will be used, which usually reduces the picture's resolution due to the low-pass filtering effect in interpolation.

Fortunately, the above transformations in the registration procedure can be cascaded. Each transformation will only result in a new sampling structure defined on the original continuous image. The interpolation from the original discrete image will be applied only when it is necessary, for example when evaluating $\mathcal{A}_{\hat{\boldsymbol{t}}_2}(\mathcal{W}_{\hat{f}_2}I_2)(\boldsymbol{x}) = (\mathcal{H}I_2)(\mathcal{S}_{\hat{\boldsymbol{t}}_2}(\mathcal{U}_{\hat{f}_2}^{-1}\boldsymbol{x}))$. Because the registration is implemented based on the positions of matching features, no error due to re-sampling will affect the registration precision.

We use $\mathcal{W}_{\hat{f}_1}I_1 \oplus \mathcal{A}_{\hat{\boldsymbol{t}}_2}(\mathcal{W}_{\hat{f}_2}I_2)$ to represent the image resulting from stitching $\mathcal{W}_{\hat{f}_1}I_1$ and $\mathcal{A}_{\hat{\boldsymbol{t}}_2}(\mathcal{W}_{\hat{f}_2}I_2)$. In the applications of obtaining large FOV pictures by stitching two adjacent views, the final de-warping process can also be cascaded as one of the sampling structure changes. The de-warping processing can be represented by $\mathcal{W}_{\hat{f}_{12}}^{-1}(\mathcal{W}_{\hat{f}_1}I_1 \oplus \mathcal{A}_{\hat{\boldsymbol{t}}_2}(\mathcal{W}_{\hat{f}_2}I_2))$ and we can see

$$\mathcal{W}_{\hat{f}_{12}}^{-1}(\mathcal{W}_{\hat{f}_1}I_1 \oplus \mathcal{A}_{\hat{\boldsymbol{t}}_2}(\mathcal{W}_{\hat{f}_2}I_2)) = \mathcal{W}_{\hat{f}_{12}}^{-1}(\mathcal{W}_{\hat{f}_1}I_1) \oplus \mathcal{W}_{\hat{f}_{12}}^{-1}(\mathcal{A}_{\hat{\boldsymbol{t}}_2}(\mathcal{W}_{\hat{f}_2}I_2)) \tag{3.15}$$

since the same focal length $\hat{f}_{12} = (\hat{f}_1 + \hat{f}_2)/2$ is used and the same reference point is chosen, namely the center of the final stitched image which can be calculated before stitching is actually carried out.

The views from panoramas can be obtained in a similar way. The reference center for de-warping and a focal length have to be specified for a segmented view from the panorama in order to implement de-warping.

The proposed method for overlap area registration based on matching feature positions can save a large number of computations since the number of matching features is much lower than the number of pixels in the overlap area. Once the matching features are obtained, the amount of computation in the proposed optimization model to find optimal affine parameters and focal length is negligible when using the matching-feature approach compared with using the texture-based approach. In addition, the cost functions of the optimization models in the matching-feature approach are directly related to the positions of the matching features. In the texture-based approach, the sampling structures of the images are first changed through the applied transformations and then required texture information is obtained from interpolation to evaluate the cost functions. As a consequence, the texture-based approach requires more additional computations.

In the feature-based approach, there are extra computations in the feature detection and matching, which serve as overhead computations in the overall computation amount. The amount of computation for feature detection and matching is different from method to method. Usually, the searching for matching features is the most expensive step. In the proposed method, the overlap areas are quite similar and thus the searching windows can be set to relatively small sizes, which greatly reduces the computation. In addition, a great deal of research, such as that in [89], is focusing on robust algorithms with low cost. As a consequence, the overhead computations can also be much lower compared with the optimizations in the texture-based approach.

3.6 Generating panoramic views

One of the most important applications for view mosaics is to generate panoramic views for IBR applications. In order to generate a cylindrical panorama, we need to stitch several images into one. Assume that images I_1, I_2, I_3, ..., I_M are taken in different directions by a camera mounted on a tripod and rotated clockwise roughly around the camera's center. The pre-captured images are overlapped with each other, and cover a 360° view. Thus the right part of I_M overlaps with the left part of I_1.

Usually, the overlap area in the images I_1 and I_2 is first registered through some transformations, such as the proposed algorithm. The transformed images are $I_1^{'}$ and $I_2^{'}$. $I_{1,2}^{'}$ represents the resulting image after stitching images $I_1^{'}$ and $I_2^{'}$. Then the overlap area between the images $I_{1,2}^{'}$ and I_3 is registered, while transforming I_3 to $I_3^{'}$. Image $I_{1,2,3}^{'}$ is obtained after stitching $I_{1,2}^{'}$ and $I_3^{'}$. Similar procedures are carried out until the transformed image $I_M^{'}$ has been stitched and the stitched image $I_{1,2,...,M}^{'}$ is obtained. Finally, the left side of image $I_{1,2,...,M}^{'}$ has to be registered and stitched with its right side in the overlap area to generate the full panorama. However, the differences between these two parts in the overlap area are usually very large, due to the accumulated registration errors, making registration very difficult. The situation is illustrated in Fig. 3.2, where we see that the transformation between I_M and I'_M is

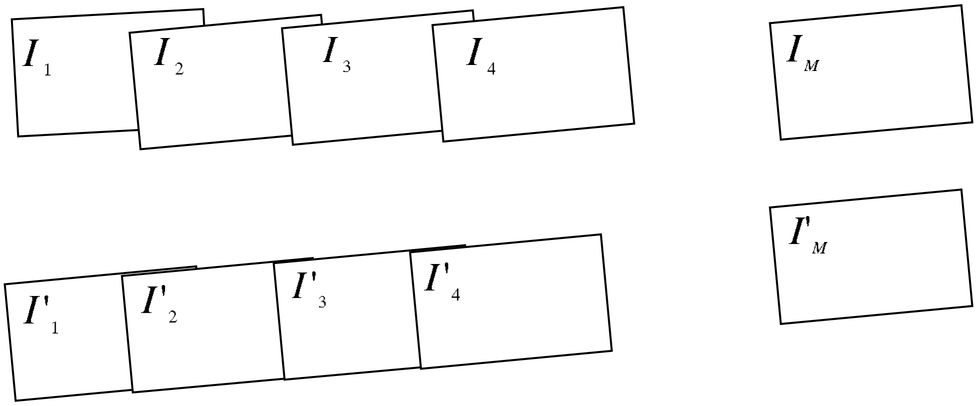


Figure 3.2: Illustration of the effect of accumulated errors in 360° panoramic view generation

very large due to the effect of accumulated errors, which brings significant changes on image I'_M. This makes the left side of I'_1 and right side of image I'_M in the overlap area very different, and the registration of these two sides in the overlap area can be very difficult. The registration errors usually remain very large. In [40], the accumulated global rotation angle is assigned to the transformations on I_1, I_2, ..., I_M after finding this global rotation angle through an initial tentative registration. Then registrations and stitching are carried out again considering the assigned rotation angles in each transformation. This will affect the registration results and the way to assign the accumulated global rotation angle on each transformation is tentative. Several trials may be required to experimentally approach a good result. A global optimization is essentially required to minimize the overall overlap-area registration errors. However, the algorithm to obtain the solution of the global optimization may be very complex.

One observation associated with this kind of situation on panoramic view generation is that every adjacent image pair with overlap usually can be registered very well individually. Based on this observation, a novel algorithm to approach the optimal global registration results in an iterative fashion is proposed. In order to describe the proposed algorithm, the overlap area registration problem is first formulated as a coordinate system conversion.

We associate a 2D coordinate system G_i with each pre-captured image I_i ($i = 1, 2, ..., M$). Without loss of generality, the origin of each 2D coordinate system is located at the left-top corner of the corresponding image. On the top of Fig. 3.3, the first three pre-captured images and the coordinate systems associated with them are illustrated. In order to generate the panorama, the ideal transformations convert each G_i into $G_i^{'}$ so that there are only horizontal offsets between any two $G_i^{'}$. The bottom of Fig. 3.3 shows the ideal relationship between $G_i^{'}$ for the first three pre-captured images after registration. Assume $\mathcal{M}_i$ represents the transformation defined on image I_i which transfers the coordinates of points in G_i to their new coordinates in $G_i^{'}$. For a scene point P, assuming the coordinate of its imaging point ${}^{i}p$ in G_i is ${}^{i}\boldsymbol{x}$, the coordinate of its imaging point ${}^{i}p^{'}$ in $G_i^{'}$ is $\mathcal{M}_i({}^{i}\boldsymbol{x})$. For the proposed registration algorithm, $\mathcal{M}_i({}^{i}\boldsymbol{x}) = \mathcal{S}_{\boldsymbol{t}_i}^{-1}(\mathcal{U}_{f_i}^{-1}({}^{i}\boldsymbol{x}))$ with affine parameter vector $\boldsymbol{t}_i$ and parameter f_i for focal length adjustment defined on I_i.

Consequently, the overlap area registration problem is equivalent to a coordinate system conversion. Suppose I_i and I_j are two adjacent images (where $j = i + 1 \mod M$). Two sets of matching feature points ${}^{i,j}\mathrm{MP}_i = \{{}^{i}\boldsymbol{x}_{(i,j),n} | n = 1, 2, ..., N_{i,j}\}$ and ${}^{i,j}\mathrm{MP}_j = \{{}^{j}\boldsymbol{x}({}^{i}\boldsymbol{x}_{(i,j),n}) | n = 1, 2, ..., N_{i,j}\}$ are in the coordinate systems G_i (in image I_i) and G_j (in image I_j), respectively. ${}^{i,j}\mathrm{MP}_i$ denotes the set of matching features between I_i and I_j and ${}^{i}\boldsymbol{x}_{(i,j),n}$ denotes the coordinate in I_i (G_i coordinate system) of one of these features. Its corresponding matching feature in I_j is at position ${}^{j}\boldsymbol{x}({}^{i}\boldsymbol{x}_{(i,j),n})$ in the coordinate system G_j. The total number of the matching features between I_i and I_j is $N_{i,j}$. The ideal transformations $\mathcal{M}_i$ and $\mathcal{M}_j$ are defined such that,

$$\mathcal{M}_i({}^{i}\boldsymbol{x}_{(i,j),n}) = \mathcal{M}_j({}^{j}\boldsymbol{x}({}^{i}\boldsymbol{x}_{(i,j),n})) + {}^{i,j}\boldsymbol{x}_0 \tag{3.16}$$

on all matching features. ${}^{i}\boldsymbol{x}_{(i,j),n}$ denotes the coordinate of one matching feature between I_i and I_j in G_i with its corresponding matching feature at position ${}^{j}\boldsymbol{x}({}^{i}\boldsymbol{x}_{(i,j),n})$ in G_j. ${}^{i,j}\boldsymbol{x}_0 = [{}^{i,j}x_0, 0]$ denotes the horizontal translation between $G_i^{'}$ and $G_j^{'}$. In practice, $\mathcal{M}_i$, $\mathcal{M}_j$ and ${}^{i,j}\boldsymbol{x}_0$ are found by minimizing $\sum_{n=1}^{N_{i,j}} |\mathcal{M}_i({}^{i}\boldsymbol{x}_{(i,j),n}) - \mathcal{M}_j({}^{j}\boldsymbol{x}({}^{i}\boldsymbol{x}_{(i,j),n})) -$

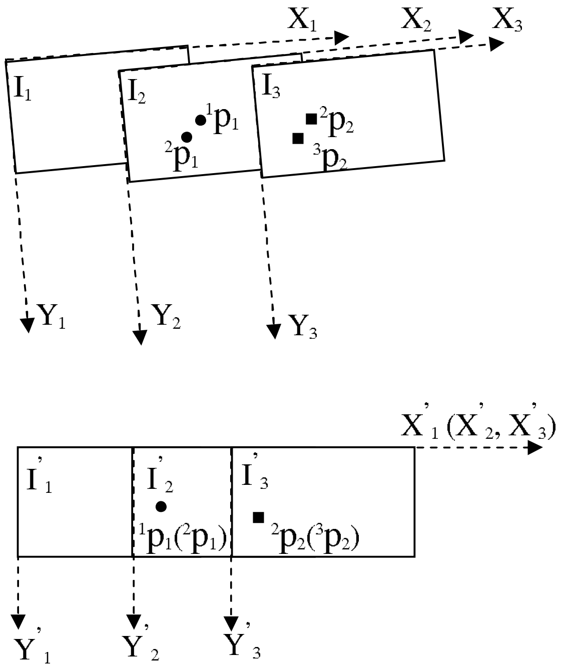


Figure 3.3: Illustration of new local minimization model

${}^{i,j}\boldsymbol{x}_0|^2$, such as in the proposed algorithm to register two adjacent overlapped images. In order to generate the panoramas, the above minimization has to be carried out on all pairs of adjacent overlapped images. The objective function for overall optimization is $\sum_{i=1}^{M}(\sum_{n=1}^{N_{i,j}} |\mathcal{M}_i({}^i\boldsymbol{x}_{(i,j),n}) - \mathcal{M}_j({}^j\boldsymbol{x}({}^i\boldsymbol{x}_{(i,j),n})) - {}^{i,j}\boldsymbol{x}_0|^2)$. A very complex global optimization algorithm might be required to solve this problem.

An efficient algorithm is proposed to solve the above optimization problem iteratively. A set of local optimizations are carried out to approach the solution of the global problem. One local optimization can be illustrated using Fig. 3.3. In the following description, the goal is to obtain the optimal transformation defined on I_j (where $j = 2$ in Fig. 3.3). Instead of searching for the optimal transformation applied on I_j to only match the overlap area between image I_i ($i = j - 1 \mod M$, where $i = 1$ in Fig. 3) and I_j, the optimal transformations applied on I_j are defined to minimize the registration errors both between I_i and I_j and between I_j and I_k ($k = j + 1 \mod M$, where $k = 3$ in Fig. 3.3). In the proposed feature matching approach, the matching features on both left and right sides of I_j are matched with their correspondences in the adjacent images, I_i and I_k. For example, on the top of Fig. 3.3, ${}^{1}p_1$ and ${}^{2}p_1$ are a pair of matching features between I_1 and I_2 on the left side of I_2 and ${}^{2}p_2$ and ${}^{3}p_2$ are a pair of matching features between I_2 and I_3 on the right side of I_2. Ideally, for all matching features, both between I_i and I_j and between I_j and I_k, the relationships

$$\begin{aligned} \mathcal{M}_i({}^{i}\boldsymbol{x}_{(i,j),n_1}) &= \mathcal{M}_j({}^{j}\boldsymbol{x}({}^{i}\boldsymbol{x}_{(i,j),n_1})) + {}^{i,j}\boldsymbol{x}_0 \\ \mathcal{M}_j({}^{j}\boldsymbol{x}_{(j,k),n_2}) &= \mathcal{M}_k^{'}({}^{k}\boldsymbol{x}({}^{j}\boldsymbol{x}_{(j,k),n_2})) + {}^{j,k}\boldsymbol{x}_0 \end{aligned} \tag{3.17}$$

are satisfied, where ${}^{j}\boldsymbol{x}_{(j,k),n_1}$ denotes the coordinate of one matching feature between I_j and I_k in G_j with its corresponding matching feature at position ${}^{k}\boldsymbol{x}({}^{j}\boldsymbol{x}_{(j,k),n_1})$ in G_k. We are looking for the optimal transformation $\mathcal{M}_j$, and thus both $\mathcal{M}_i$ and $\mathcal{M}_k^{'}$ are temporarily fixed. $\mathcal{M}_i$ is the updated transformation applied on image I_i. $\mathcal{M}_k^{'}$ is the transformation that will be updated after we obtained the optimal $\mathcal{M}_j$ using equation (3.17). $\mathcal{M}_k^{'}$ could be the initial transformation or the optimal transformation found in the last iteration when using an iterative method. The correspondent objective function for optimization over the parameters of $\mathcal{M}_j$, ${}^{i,j}\boldsymbol{x}_0$ and ${}^{j,k}\boldsymbol{x}_0$ is

$$\begin{aligned} &\frac{1}{N_{i,j}} \sum_{n=1}^{N_{i,j}} |\mathcal{M}_i({}^{i}\boldsymbol{x}_{(i,j),n}) - \mathcal{M}_j({}^{j}\boldsymbol{x}({}^{i}\boldsymbol{x}_{(i,j),n})) - {}^{i,j}\boldsymbol{x}_0|^2 + \\ &\qquad \frac{1}{N_{j,k}} \sum_{n=1}^{N_{j,k}} |\mathcal{M}_j({}^{j}\boldsymbol{x}_{(j,k),n}) - \mathcal{M}_k^{'}({}^{k}\boldsymbol{x}({}^{j}\boldsymbol{x}_{(j,k),n})) - {}^{j,k}\boldsymbol{x}_0|^2. \end{aligned} \tag{3.18}$$

The overall procedure to generate panoramas using the matching-feature approach is as follows:

1) Apply warping with initial focal length f_0 on all pre-captured images I_1, I_2, ..., I_M. These are the initial transformations $\mathcal{M}_1^{'}$, $\mathcal{M}_2^{'}$, ... $\mathcal{M}_M^{'}$; set $g = 0$.

2) Set $j = 2$ and $\mathcal{M}_1 = \mathcal{M}_1^{'}$.

3) Search for the best transformation $\mathcal{M}_j$ (including both affine transformation and focal length adjustment) for I_j by minimizing equation (3.15).

4) $j = j + 1 \mod M$.

5) $j = 2$? if not, go to step 3); otherwise, go to step 6).

6) $g = g + 1$; stop? if not, set $\mathcal{M}_1^{'} = \mathcal{M}_1$, $\mathcal{M}_2^{'} = \mathcal{M}_2$, ..., $\mathcal{M}_M^{'} = \mathcal{M}_M$ go to step 2).

The condition to terminate the iteration can be: (1) the registration errors are small enough compared to a predefined threshold; or (2) the change of the registration errors is small enough compared to a predefined threshold; or (3) the number of the overall iterations g has reached a predefined value. In our experiments, we used g to terminate the iteration. Good results can be obtained by three to five overall iterations. The techniques of registration and stitching on a strip area can be incorporated in the above algorithm in a straightforward manner.

3.7 Experimental results

In this section, we first test the proposed algorithm for the stitching of two adjacent images with some overlap area. In this test, a pair of adjacent images is stitched to synthesize a view with a larger FOV. Then, the results are shown when stitching two registered adjacent images on stitching areas with different widths. The registration is implemented using both the traditional approach (note: by traditional approach, we mean the algorithms using equation (3.2) for registration and blending on a wide overlapped area) and the proposed approach for comparison. The mean square errors (MSE) between the positions of matching features in the overlap areas (equivalent to the obtained minimum in equations (3.4), (3.9), etc.) will be used to evaluate the registration results. Using the changes of these MSE values, the contributions from different adjustments that have been presented can be illustrated. Then, the proposed multi-resolution stitching algorithm will be tested. Finally, the proposed algorithm for generation of 360° panoramas is implemented as one application for view mosaicking. In all experiments, the original images were captured by a digital camera mounted and rotated on a tripod, at different rotation angles. The camera poses were just roughly adjusted to be levelled by hand. The image size for all pre-captured images is 1024 pixels in width and 768 pixels in height. *Informal subjective evaluation* will be used to observe and compare the artifacts, the discontinuities, or the intensity changes in the stitched images. This is adequate because these artifacts, discontinuities, or intensity changes are very obvious.

Fig. 3.4 shows the results obtained in experiment 1. The overlap images $I_{1\text{com}}$ and $I_{2\text{com}}$ with detected Harris corners are shown in Fig. 3.4 (a), (b). The geometric relationships between the matching features after registration using equation (3.2) are shown in Fig. 3.4 (c). The geometric relationships between these matching features after overlap area registration using the proposed algorithm are shown in Fig. 3.4 (d). If we simply warp the images with the initial focal length and adjust the two images through shifting along horizontal and vertical directions (traditional registration approach), the stitched results are shown in Fig. 3.4 (e). The stitching was carried out in the indicated strip area. Fig. 3.4 (f) shows the stitching result after applying the proposed registration algorithm and blending on the indicated strip block. The blending is carried out in the area between the two vertical lines in each stitched image. It is clear that the match of the two image at the overlap area is much better and that the stitched view is significantly improved with the proposed registration algorithm (for example as the edges of the shelves near the bottom).

Fig. 3.4 (g) shows the stitching result using the proposed approach but blending on a wide overlapped area in Experiment 1. Although the matching between two images at the entire overlapped area is much improved and there is no significant mismatched place (which may show obvious “ghost image”), we can notice that the image in Fig. 3.4 (g) looks blurred compared to Fig. 3.4 (f). This is due to the blending of the overlap area with generally small mismatching. The small mismatching will not bring significant difference (or disparities) between the overlapped areas before stitching. Thus there is no obvious “ghost image” appearing in the stitched image but the image quality will be generally degraded. For the sake of comparison, Fig. 3.4 (h) shows the stitching result with traditional registration approach and blending on a wide overlapped area. This is the worst case scenario. We can notice that there are lots of significantly mismatched places which bring the “ghost image” in the resulting images (for example, in the lower and central area of the image).

As a conclusion, the image Fig. 3.4 (f) gives the best stitching result. In this experiment, we can notice that there is a possible faulty matching feature at the tops of the images, which are low texture areas. However, it does not significantly affect the proposed registration algorithm and the stitching result because the number of matching features in the low texture area is very small.

Experiment 2 was carried out on a pair of images intentionally chosen to have large depth variations. Due to the large depth variations for these two adjacent images, it is very difficult to get a good registration result in the overlap area. The overlap images are shown in Fig. 3.5 (a), (b). Fig. 3.5 (c) shows the stitching result using traditional registration approach and blending in a wide area, with the stitching result using a traditional registration approach but blending in a strip area as shown in Fig. 3.5 (d). Fig. 3.5 (e) shows the stitching result after registration using the proposed method but blending in a wide area, with the stitching result after registration using proposed method and blending on a strip area shown in Fig. 3.5(f). We can see that the proposed method can give better registration results and the technique of stitching on a strip area can significantly improve the visual quality of the stitched images.

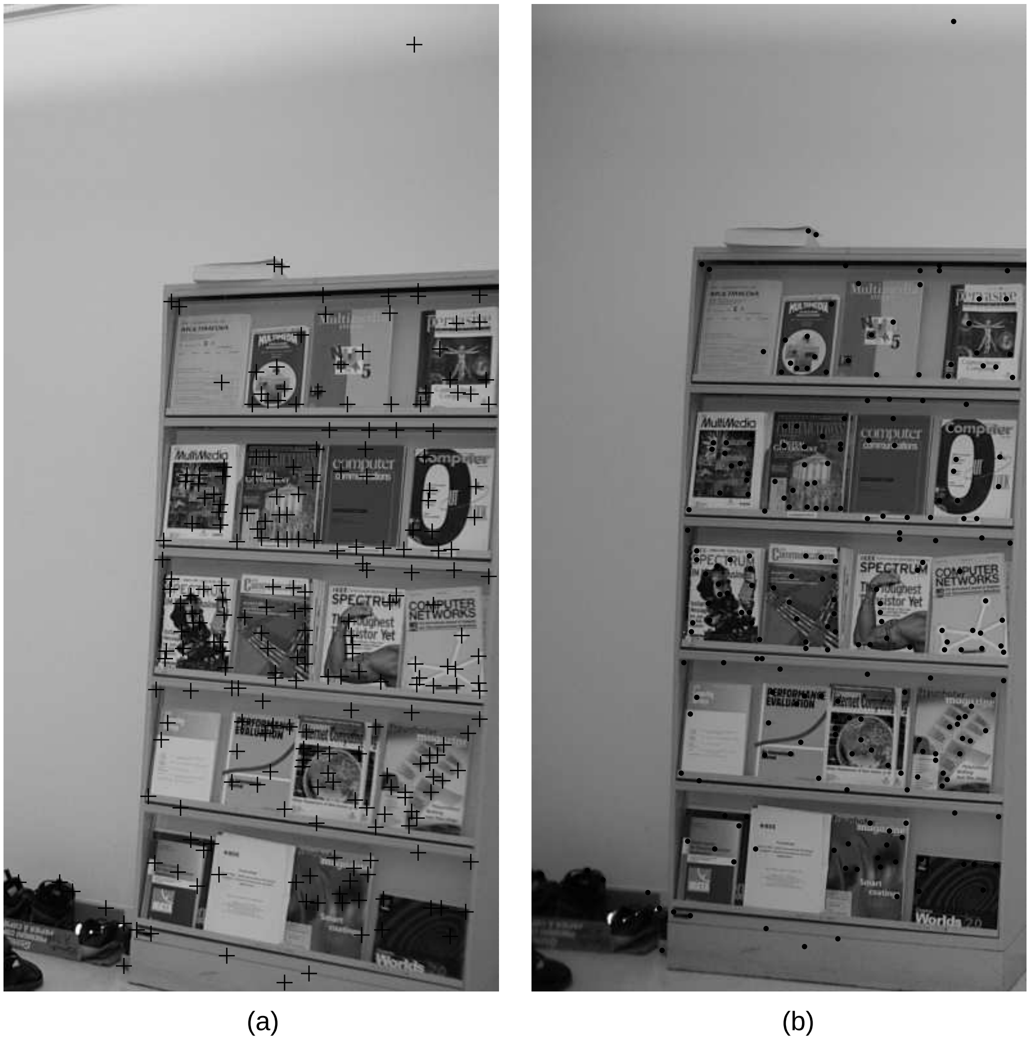


Figure 3.4: Results for experiment 1. (a) The matching features in $I_{1\mathrm{com}}$. (b) The matching features in $I_{2\mathrm{com}}$.

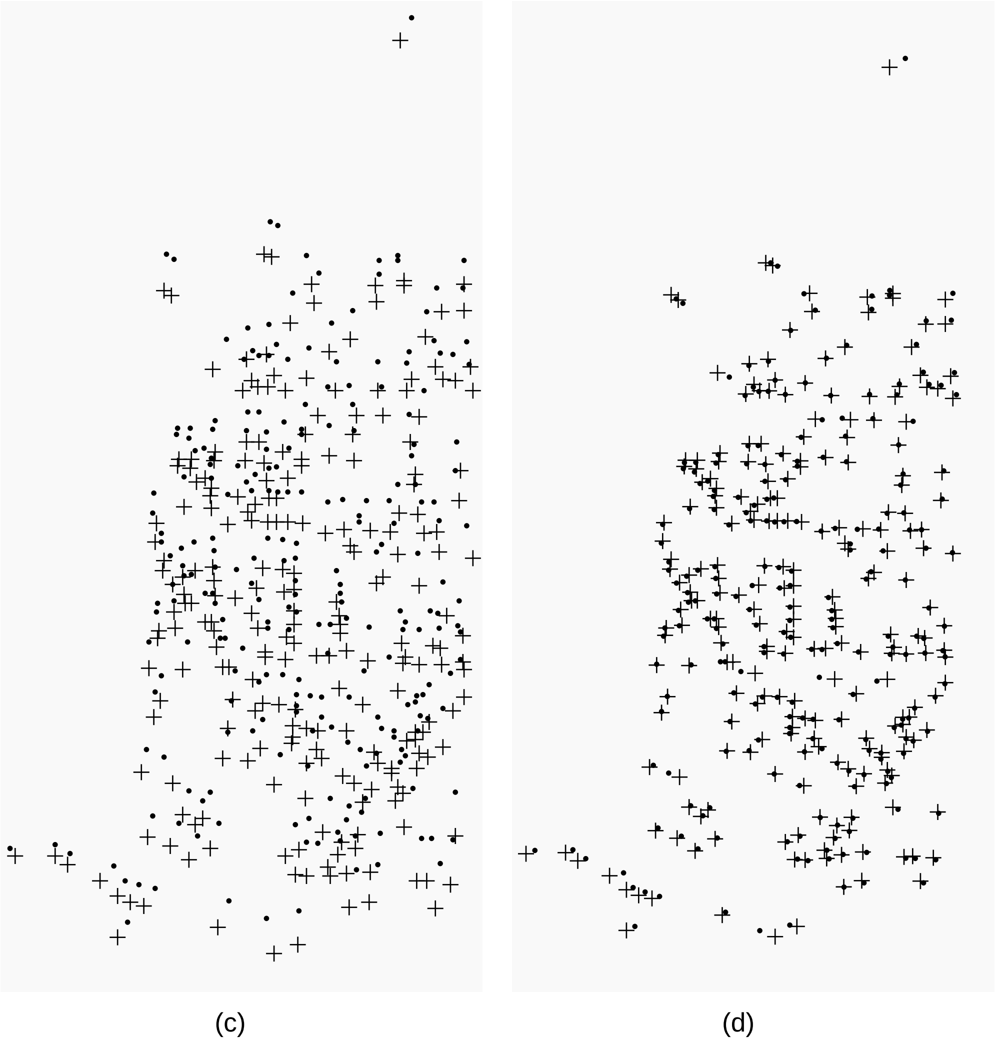


Figure 3.4 Results for experiment 1 (continued). (c) The relationships between matching features after registration using the traditional approach. (d)The relationships between matching features after registration using the proposed algorithm.

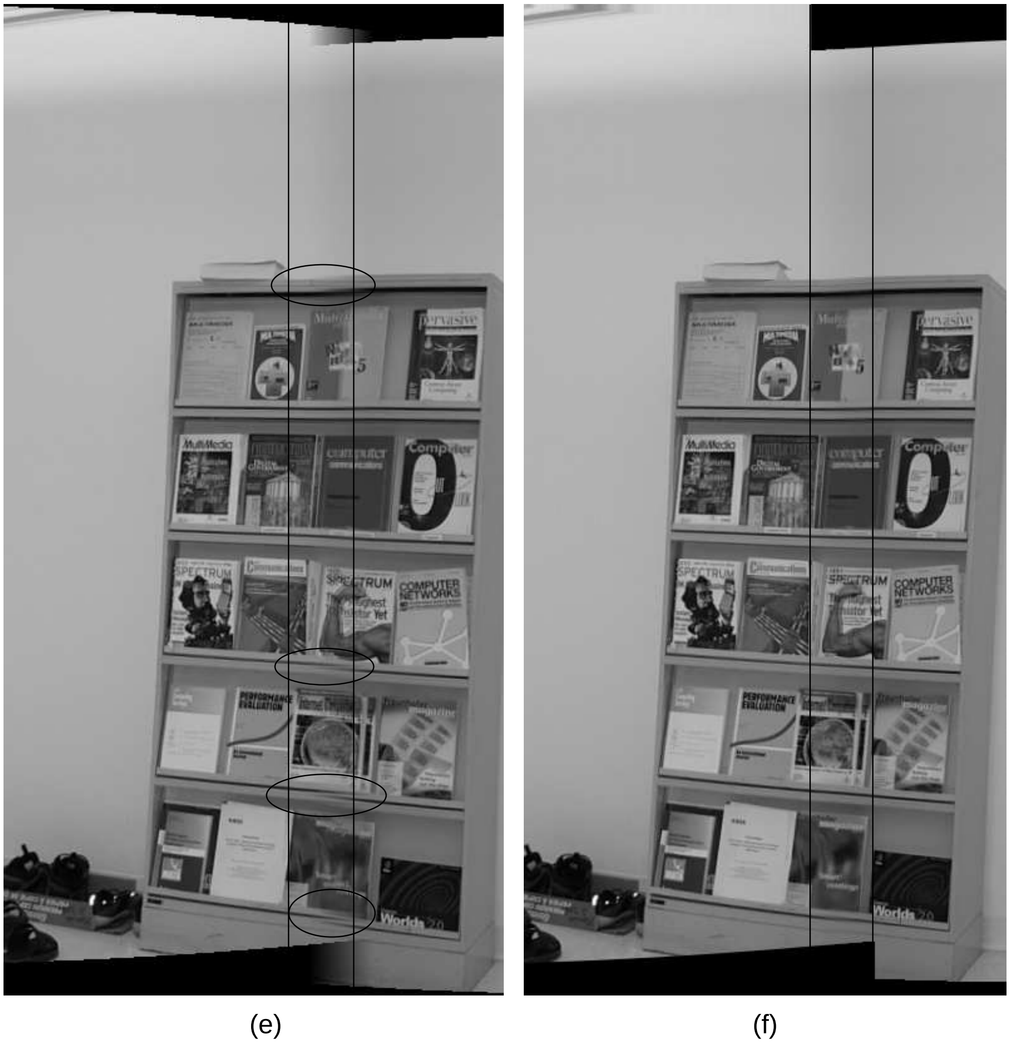


Figure 3.4 Results for experiment 1 (continued). (e) Stitching result with traditional registration approach but blending on a strip block. (f) Stitching result after registration using proposed method and blending on a strip block.

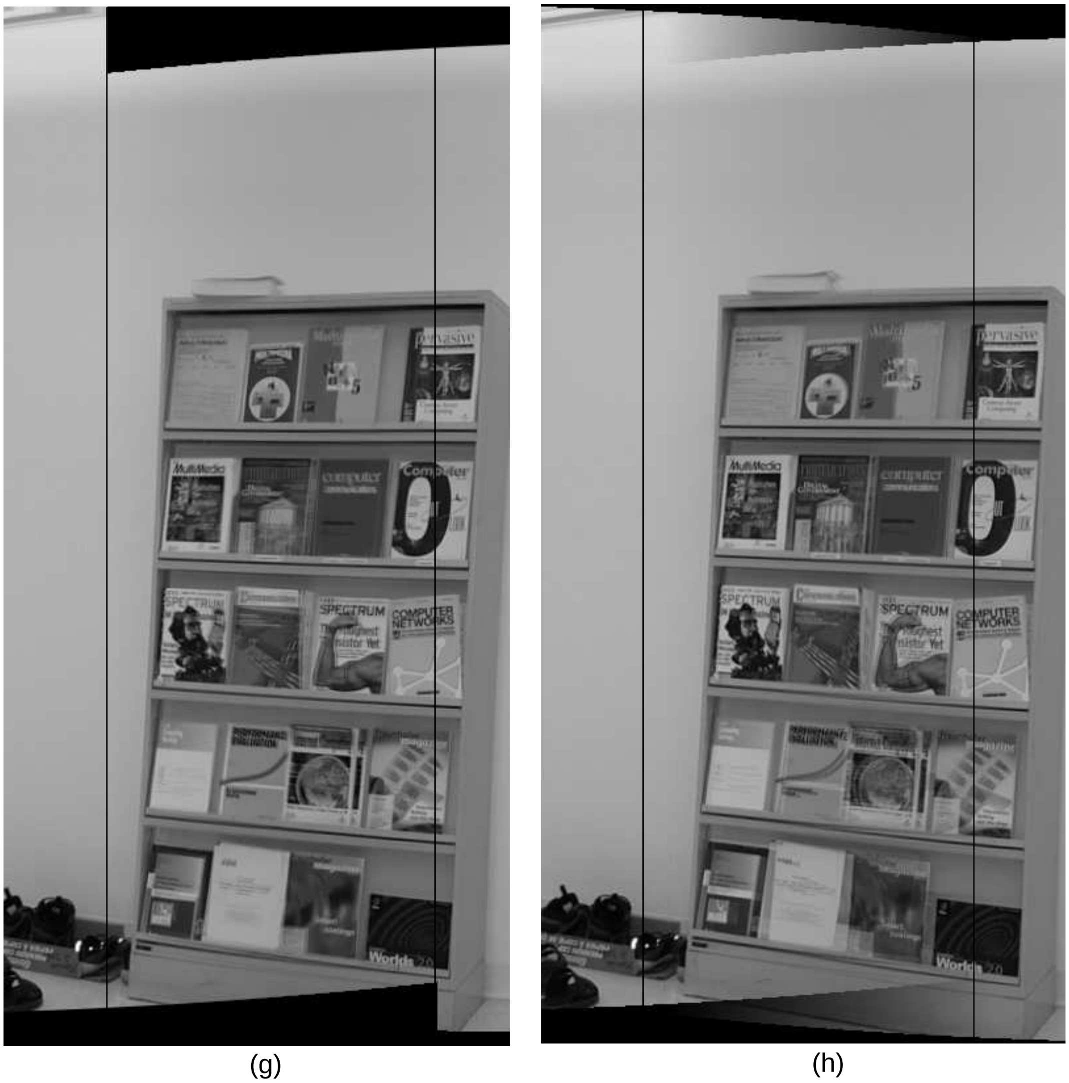
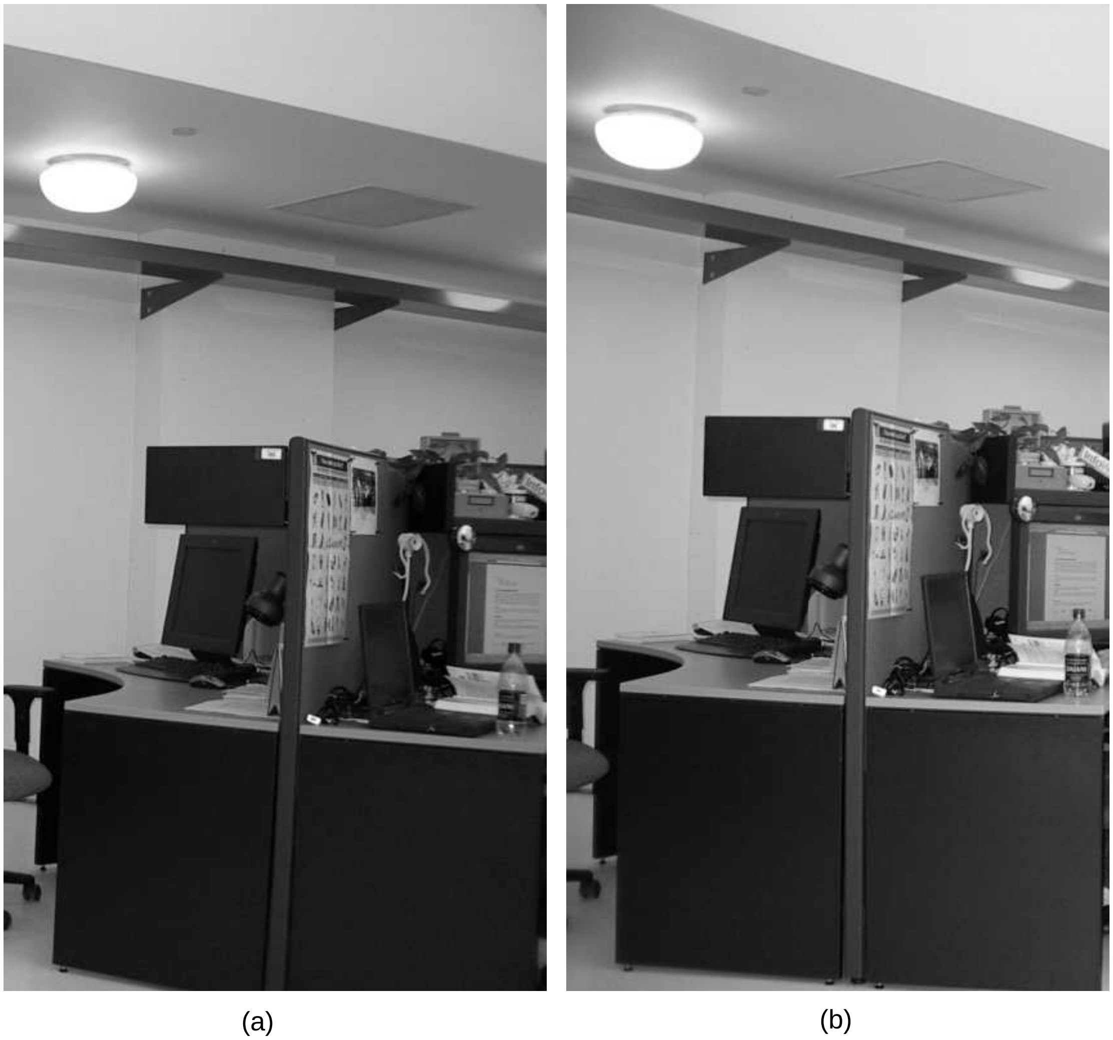


Figure 3.4 Results for experiment 1 (continued). (g) Stitching result with the proposed registration algorithm but blending on a wide overlapped area. (h) Stitching result after registration using traditional method and blending on a wide overlapped area.



(a)

(b)

Figure 3.5: Results for experiment 2. (a) $I_{1\text{com}}$. (b) $I_{2\text{com}}$.

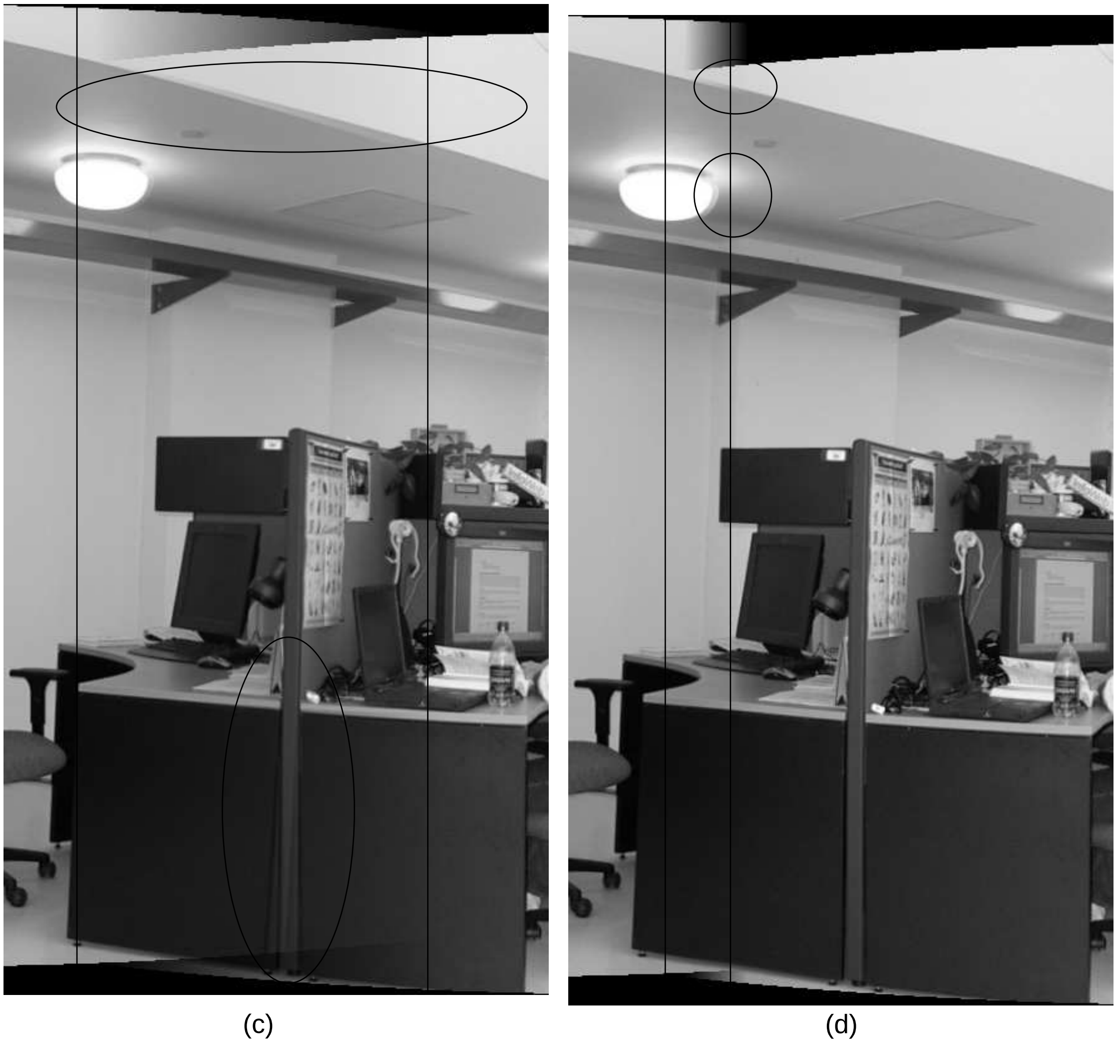


Figure 3.5 Results for experiment 2. (c) Stitching result with traditional registration approach and blending in a wide area. (d) Stitching result using traditional registration approach but blending in a strip area.

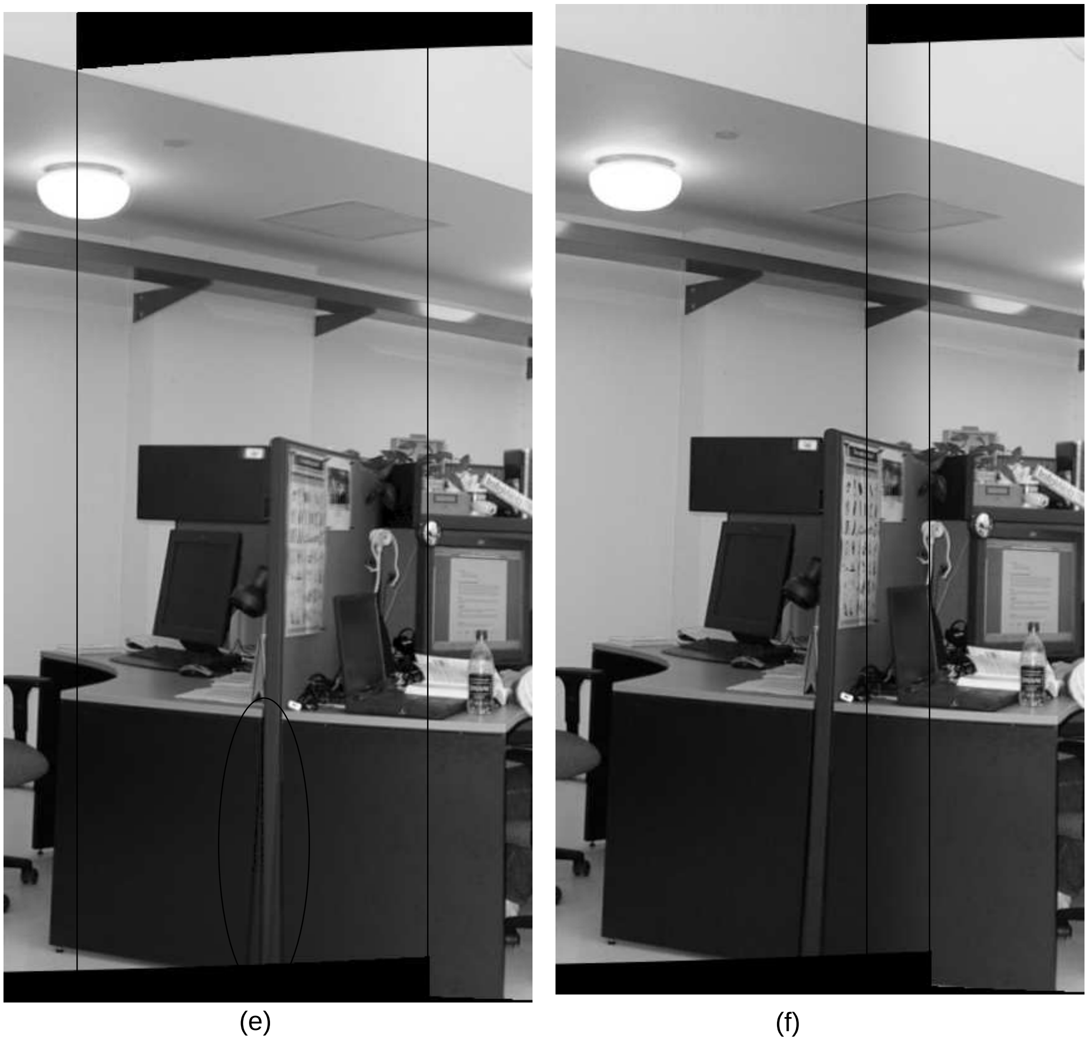


Figure 3.5 Results for experiment 2. (e) Stitching result after registration using proposed method but blending in a wide area. (f) Stitching result after registration using proposed method and blending on a strip area.

Table 3.1: MSE between the positions of matching features in Experiment 1 and 2. (1) After relative shifting. (2) After initial warping and relative shifting. (3) After applying affine adjustment on I_1. (4) After applying focal-length adjustment on I_1. (5) After applying focal-length adjustment on I_2.

Experiment	(1)	(2)	(3)	(4)	(5)
1	0.3364	0.3063	0.2182	0.2176	0.2158
2	1.1798	0.6034	0.5872	0.5743	0.5675

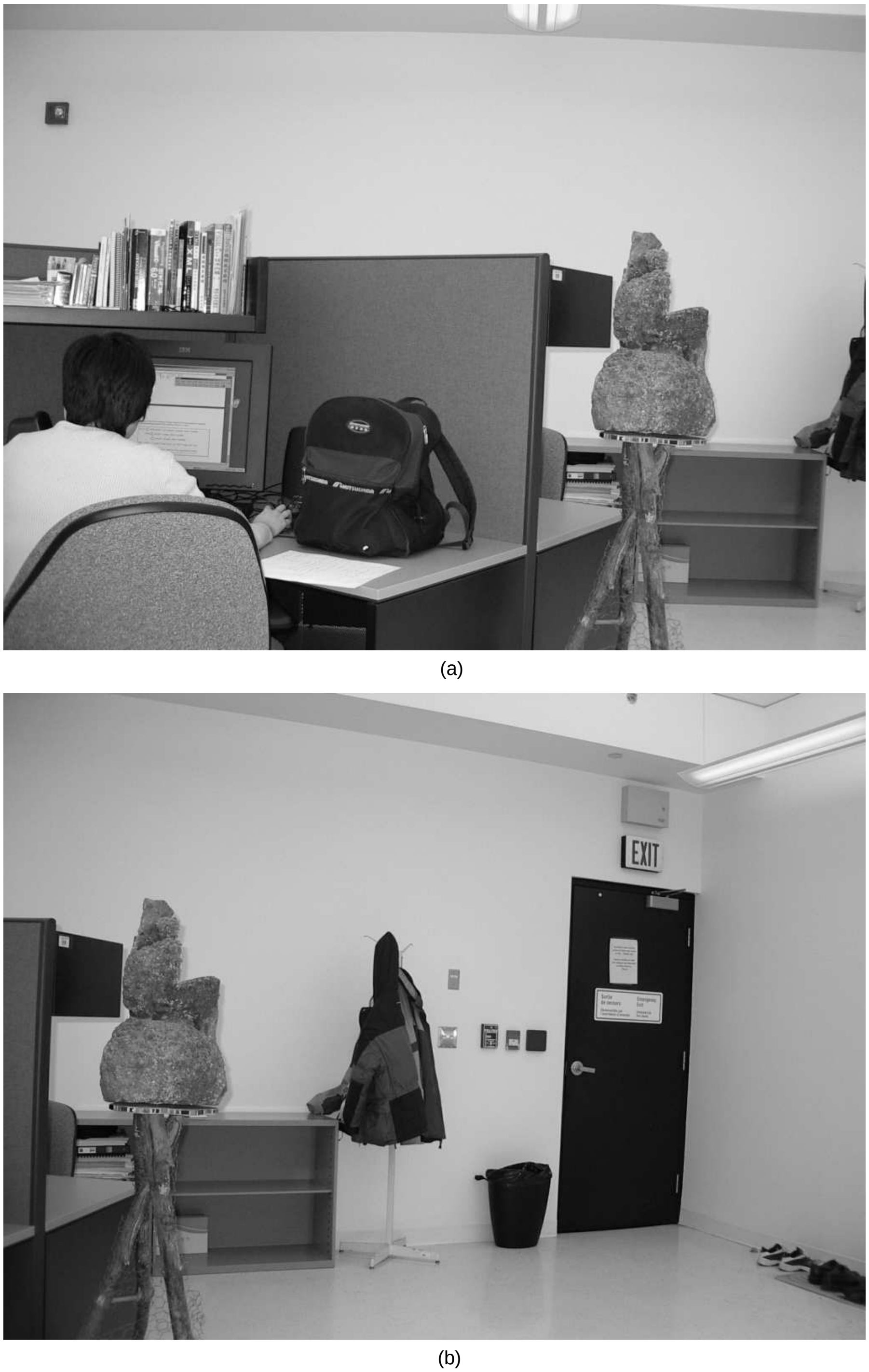
Table 3.1 shows the MSE between the positions of matching features during the registrations in Experiment 1 and 2. The values were calculated based on the same set of matching features in the overlap area in each experiment, respectively. The data in column (1) are the minimal MSE values between the positions of the matching features after applying the traditional registration approach (equation (3.2)). The data in column (2) show minimal MSE values between the positions of the matching features after initial warping and relative shifting of two adjacent images along horizontal and vertical directions. The data in column (3) show the MSE between the positions of the matching features after affine adjustment. The data in column (4) and (5) show the MSE between the positions of the matching features after focal length adjustments on two adjacent images, respectively. Because the widths of overlap areas, distributions and numbers of detected matching features are different in Experiment 1 and 2, the MSE values between Experiment 1 and 2 are not comparable.

The multi-resolution-based stitching algorithm was tested in Experiment 3. Two adjacent images with large luminance change are shown in Fig. 3.6 (a) and (b). If these two images are stitched on a narrow strip block using the proposed method, the significant luminance change can be observed in the resulted image. Fig. 3.6 (c) shows the stitched overlapped area.

Using the proposed multi-resolution approach, one pair of frequency-complementary filters (one high pass filter and one low pass filter) are designed. Each original image is decomposed into two sub-band images. Then, the low resolution images are stitched on a wide overlapped area while high resolution images are stitched in a narrow strip area. Then two stitched images with different resolutions are composed to generate the final stitched image. As a consequence, the rapid luminance change doesn't exist anymore. Fig. 3.6 (d) shows the stitched overlapped area.

In the last experiment, we tested the proposed registration algorithm in generating 360° panoramas. A tentative panoramic view is generated using the traditional approach in order to check the imaging conditions of the pre-captured images. The procedure was described in Section (3.6) with $M = 12$. We found that there is a huge vertical offset (around 200 pixels) between the left side and right side of the stitched images $I'_{1,2,\ldots,12}$ due to the accumulated errors. A simple global rotation introduced severe distortion, bringing significant difference between the left side and right side of $I'_{1,2,\ldots,12}$ in the overlap area.

The proposed algorithm for global registration in generating panoramas was implemented and the final stitching results on the overlap area between the first image and the last image are shown in Fig. 3.7. Fig. 3.7 (b) shows the reference view, or ground truth, from the original view. Fig. 3.7(b), Fig. 3.7(c), and Fig. 3.7(d) show the stitching results after one iteration, two iterations and three iterations, respectively. The results show that the proposed algorithm converges very quickly in obtaining an optimized solution. Fig. 3.8 shows two final panoramas generated using the proposed algorithm for different scenarios (indoor and outdoor).



(a)

(b)

Figure 3.6: The source images: (a) left image (b) right image

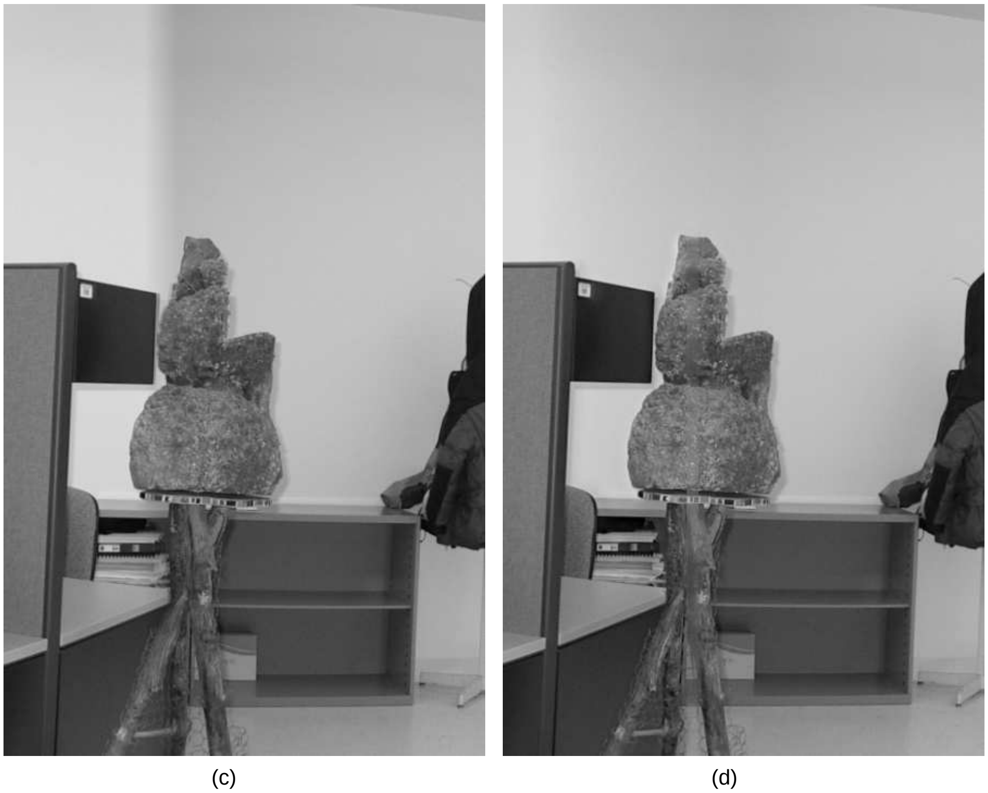


Figure 3.6 The stitched image on the overlapped area: (c) stitching on a strip block (d) stitching on a strip block using the proposed multi-resolution-based approach

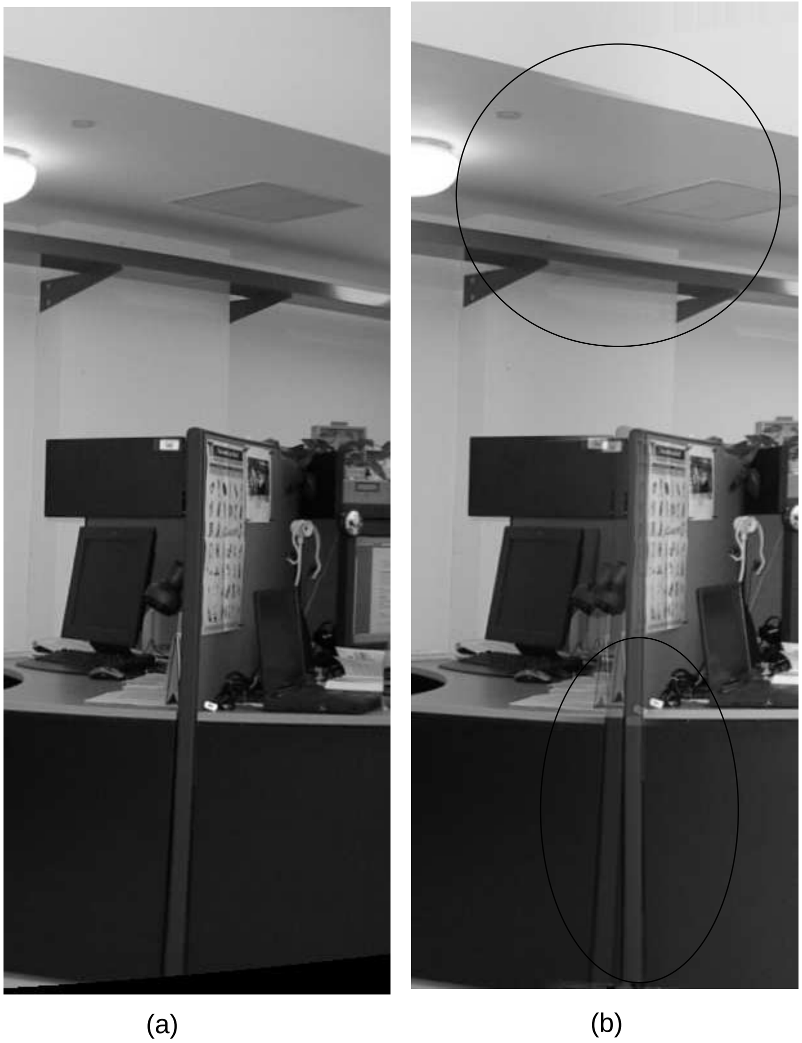


Figure 3.7: The stitching results in the overlap area between the first image and the last image in generating panoramas with different iterations using proposed algorithm: (a) reference view (ground truth) (b) after 1 iteration

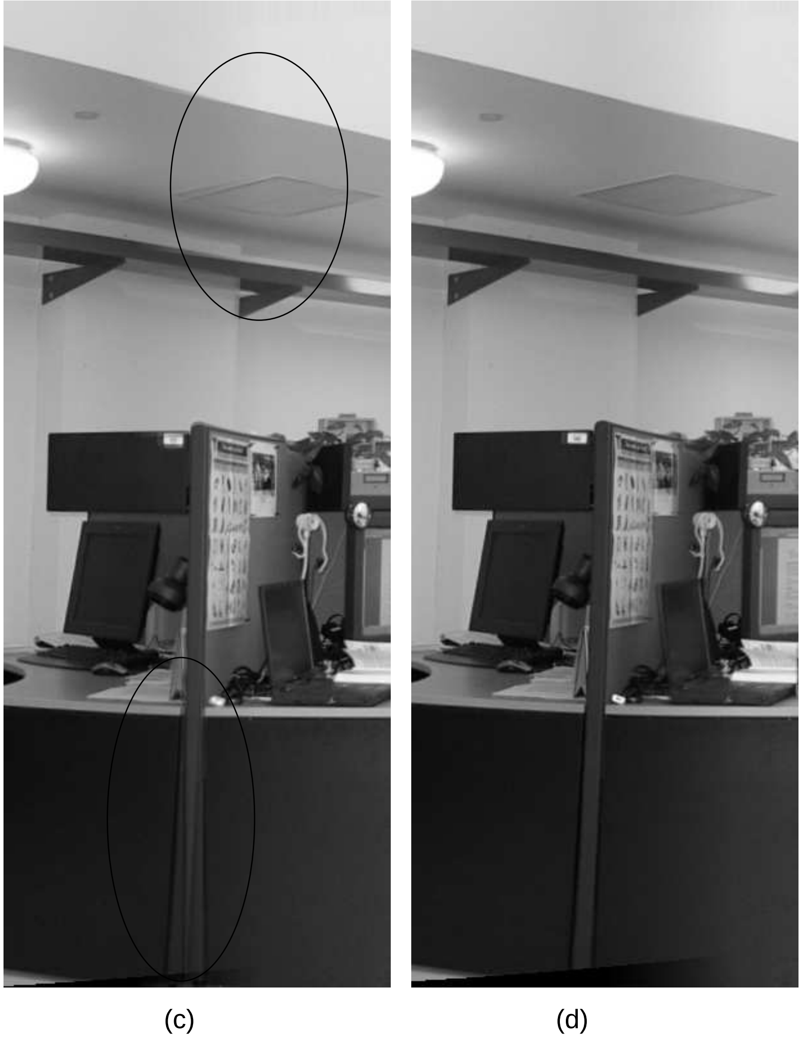


Figure 3.7 The stitching results in the overlap area between the first image and the last image in generating panoramas with different iterations using proposed algorithm: (c) after 2 iterations (d) after 3 iterations

3.8 Summary

This chapter has proposed an overlap-area registration technique for view mosaicking using an affine transformation adjustment together with a nonlinear refinement through focal length adjustment. A feature-based overlap-area registration algorithm has been described for implementation of the proposed method. The principle of using matching feature points as control points for view transformations can be applied more broadly. One major advantage of using matching feature points as control points is that the amount of computation can be greatly reduced. Moreover, the strategy of registration and stitching on a strip block within the overlap area can significantly improve the visual quality of the mosaicked images.

In our approach, the registration of the adjacent images was formulated as a general coordinate-system transformation module which is independent of the particular registration methods. The framework for generating panoramas through adjacent image registration can then be clearly illustrated. Due to the complexity in obtaining a global optimization solution for the registration errors in the overlap areas of all pre-captured images, a novel simple iterative algorithm was proposed with the experimental results showing that the results converge to a good solution very quickly.

In addition, we observe that these view transformations can be considered as sampling structure changes which can be combined into one step in order to avoid unnecessary interpolation, which will generally reduce the texture resolution of the images. The principle can be applied to other applications involving multiple transformations on the images, when some or all of these transformations can be combined into a single one.

Finally, the simulation results show that the proposed algorithm can obtain panoramas 360° of good quality.

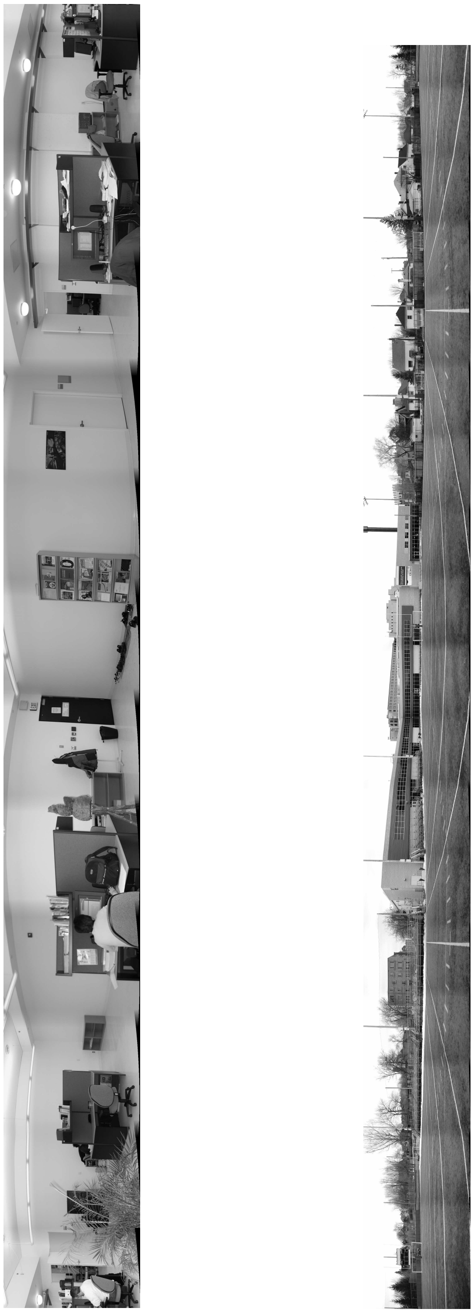


Figure 3.8: The generated panoramas using the proposed method: left (indoor scene) and right (outdoor scene)

Chapter 4

View interpolation from adjacent views

View interpolation from adjacent views is another approach for the IBR application. This approach requires dense disparity maps between the pre-captured images. It is well-known that precise dense disparity maps are very difficult to obtain in practice [90]. In the literature, view-interpolation algorithms have been developed for the applications of interpolating the intermediate views in stereo pairs or interpolating the images within a video sequence, etc. For these types of applications, the view changes are usually small and the existing algorithms in the literature for optical flow or disparity estimation can give acceptable results. However, in IBR applications, the view changes may be large and forward or backward motions of the camera may be involved.

In this chapter, we will study the methods for disparity-based view interpolation. The main topics will be dense disparity estimation for different camera-motion situations used in IBR applications. Methods will be developed to relate such camera-motion situations to the conditions in the existing methods for dense disparity estimation in the literature. A triangulation-based view interpolation method will be introduced.

Fig. 4.1 shows the general scenario where view interpolation is required. Images

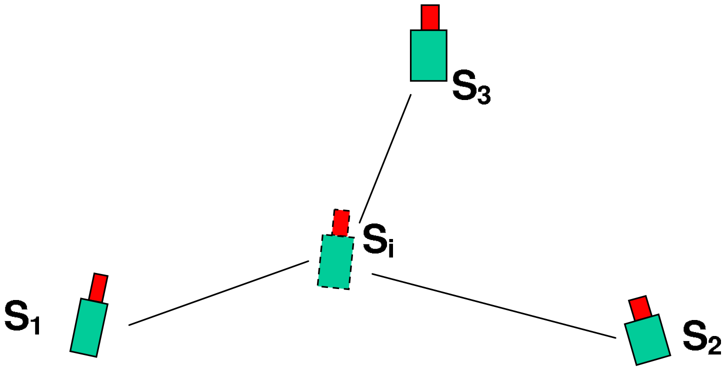


Figure 4.1: view interpolation from the adjacent pre-captured images: S_k is the position at which image I_k is taken, $k = 1, 2, 3$.

I_1, I_2, and I_3 from camera positions S_1, S_2 and S_3 with similar view directions are pre-captured. S_i is the virtual camera position where the interpolated view I_i would have to be taken if no view-interpolation were applied during navigation.

View interpolation from multiple source images has been studied in [15], [16] which is implemented by a tri-linear tensor-based approach. However, this method assumes that the dense disparities between different views are known. It is known that the precise dense disparities between multiple views are very difficult or expensive to obtain. Thus, the simulations for this method in [15], [16] were carried out in the scenario of the scenes with only a single object.

In Section 4.1, a triangulation-based view interpolation algorithm is proposed for such a general scenario. Experimental results which are based on such an algorithm will be given for both the above three-camera scenario and the special scenario where there are only two pre-captured images available. In this two-camera situation, we assume that the two cameras are parallel (or roughly parallel) with each other and are located side by side.

In Section 4.2, another special situation will be studied. In this situation, two images are captured at two positions along the camera axis with the same view direction. This is very similar to a zooming effect. This two-camera scenario simulates the camera forward/backward motion when taking pre-captured images in the direction of camera motion. No research work dealing with such camera motion scenario has been reported in the literature. The proposed view interpolation algorithm is novel.

4.1 Scenario I: A triangulation-based view interpolation algorithm

Triangulations have been used in computer graphics for texture mapping [91] and recently have been applied in view interpolation [92]. However, in [92], quasi-dense matching points have to be detected instead of matching features, and a complex rendering strategy is used.

The proposed view interpolation algorithm is based on triangulation together with affine transformation for each triangular patch instead of dense disparities. As a consequence, it requires a large number of approximately uniformly distributed feature matchings. We develop a multi-view matching detection algorithm based on a new view-matching relationship constraint, along with some other traditional epipolar constraints commonly used in computer vision.

4.1.1 A view-consistent model for IBR application

Assume that Π denotes a specified navigation area, and $I_{\boldsymbol{s}_k}$, $k = 1, 2, ..., K$, are pre-captured images with the camera projection center at positions $\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_K$. Π is a polygon that is the convex hull of $\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_K$ and $\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_K \in \Pi$. These images have similar, but not necessarily identical, viewing directions of the scene. For the IBR application, the objective is to synthesize an arbitrary view at some position $\boldsymbol{s_a} \in \Pi$ with a similar viewing direction, while satisfying the following consistency conditions:

$$I_{\boldsymbol{s_a}} = I_{\boldsymbol{s}_k}, \text{if } \boldsymbol{s_a} = \boldsymbol{s}_k, \text{for any } k \in \{1, 2, ..., K\}. \tag{4.1}$$

In a more general situation, for any of the two views $I_{\boldsymbol{s}_a}$ and $I_{\boldsymbol{s}_b}$ within Π,

$$\lim_{\boldsymbol{s_a} \to \boldsymbol{s_b}} \sum_m \sum_n |I_{\boldsymbol{s_a}}(m, n) - I_{\boldsymbol{s_b}}(m, n)| = 0. \tag{4.2}$$

where, $I_{\boldsymbol{s}_a}$ and $I_{\boldsymbol{s}_b}$ can be pre-captured or interpolated views.

Based on this view-consistent interpolation model, the synthesized virtual views may be different from the real physical ones, but will look reasonable during the navigation.

4.1.2 Interpolation model with consistent feature positions

A matching-based view interpolation algorithm is proposed. Matching features among pre-captured images are first detected and used as control points [93]. The above texture-consistent interpolation model is converted to the feature-position-consistent interpolation model.

Assume that N feature points have been found in all of the K pre-captured images and that there is a unique one-to-one corresponding relationship of these N feature points among the K pre-captured images. Further assume that the corresponding points of these N feature points will appear in any of the views in the navigation region Π that will be synthesized from these K pre-captured images.

We use the feature points $\boldsymbol{x}_{s_1,n}, (n = 1, 2, ..., N)$ in $I_{\boldsymbol{s}_1}$ as the reference points. The matchings between views $I_{\boldsymbol{s}_a}$ and $I_{\boldsymbol{s}_b}$ are represented as two sets of feature points $\mathrm{MP}_a = \{\boldsymbol{x}_{s_a}(\boldsymbol{x}_{s_1,n})|n = 1, 2, ..., N\}$ and $\mathrm{MP}_b = \{\boldsymbol{x}_{s_b}(\boldsymbol{x}_{s_1,n})|n = 1, 2, ..., N\}$ in $I_{\boldsymbol{s}_a}$ and $I_{\boldsymbol{s}_b}$, respectively. For any feature point $\boldsymbol{x}_{s_a}(\boldsymbol{x}_{s_1,n}) \in \mathrm{MP}_a$, its matching point in $I_{\boldsymbol{s}_b}$ is denoted as $\boldsymbol{x}_{s_b}(\boldsymbol{x}_{s_1,n}) \in \mathrm{MP}_b$. Then the correspondent feature-position-consistent interpolation model can be represented as

$$\boldsymbol{x}_{s_a}(\boldsymbol{x}_{s_1,n}) = \boldsymbol{x}_{s_k}(\boldsymbol{x}_{s_1,n})\, n = 1, 2, ..., N, \text{ if } \boldsymbol{s_a} = \boldsymbol{s}_k \tag{4.3}$$

for any $k \in 1, 2, ..., K$. $\boldsymbol{x}_{s_k}(\boldsymbol{x}_{s_1,n})$ denotes the matching points in the pre-captured image $I_{\boldsymbol{s}_k}$. Also, in more general situations,

$$\lim_{\boldsymbol{s_a} \to \boldsymbol{s_b}} |\boldsymbol{x}_{s_a}(\boldsymbol{x}_{s_1,n}) - \boldsymbol{x}_{s_b}(\boldsymbol{x}_{s_1,n})| = 0, n = 1, 2, ..., N. \tag{4.4}$$

It is obvious that the feature-position-consistent interpolation model will converge with the texture-consistent interpolation model when the number N of matching points becomes large enough and distributed on the entire image.

We will address the scenario that the sub-area is a triangle, and that $K = 3$. The triangular area is specified by its vertices, the positions where three pre-captured images are taken. Assume that the three pre-captured images with similar viewing directions are $I_{\boldsymbol{s}_A}$, $I_{\boldsymbol{s}_B}$, and $I_{\boldsymbol{s}_C}$, taken at positions $\boldsymbol{s}_A$, $\boldsymbol{s}_B$, and $\boldsymbol{s}_C$ (i.e., $\boldsymbol{s}_1 = \boldsymbol{s}_A$, $\boldsymbol{s}_2 = \boldsymbol{s}_B$, $\boldsymbol{s}_3 = \boldsymbol{s}_C$ according to the above representations). The matching feature points in these three images form the sets MP_A, MP_B, and MP_C. We will describe the procedure to generate an arbitrary view $I_{\boldsymbol{s}_a}$ with the similar viewing direction at position $\boldsymbol{s}_a$, which is within the triangle with vertices $\boldsymbol{s}_A$, $\boldsymbol{s}_B$ and $\boldsymbol{s}_C$.

Following the feature-position-consistent view interpolation model, the positions of the feature matchings in $I_{\boldsymbol{s}_i}$ can be calculated as

$$\boldsymbol{x}_{s_a}(\boldsymbol{x}_{s_A,n}) = \frac{(1+\eta_A)\cdot \boldsymbol{x}_{s_A,n} + \eta_B \cdot \boldsymbol{x}_{s_B}(\boldsymbol{x}_{s_A,n}) + \eta_A \cdot \eta_B \cdot \boldsymbol{x}_{s_C}(\boldsymbol{x}_{s_A,n})}{(1+\eta_A)(1+\eta_B)} \tag{4.5}$$

for $n = 1, 2, ..., N$. This is inherited from the geometric relationship

$$\boldsymbol{s}_a = \frac{(1+\eta_A)\cdot \boldsymbol{s}_A + \eta_B \cdot \boldsymbol{s}_B + \eta_A \cdot \eta_B \cdot \boldsymbol{s}_C}{(1+\eta_A)(1+\eta_B)} \tag{4.6}$$

with

$$\begin{aligned} \eta_A &= \frac{|\boldsymbol{s}_B - \boldsymbol{s}_P|}{|\boldsymbol{s}_P - \boldsymbol{s}_C|} \\ \eta_B &= \frac{|\boldsymbol{s}_i - \boldsymbol{s}_A|}{|\boldsymbol{s}_P - \boldsymbol{s}_i|} \end{aligned} \tag{4.7}$$

where, $\boldsymbol{s}_P$ is the intersection of line $\boldsymbol{s}_A\boldsymbol{s}_a$ with $\boldsymbol{s}_B\boldsymbol{s}_C$. The geometric relationships between $\boldsymbol{s}_A$, $\boldsymbol{s}_B$, $\boldsymbol{s}_C$, $\boldsymbol{s}_P$ and $\boldsymbol{s}_a$ are illustrated in Fig. 4.2.

Thus, all points $\boldsymbol{x}_{s_a}(\boldsymbol{x}_{s_A,n})$ form the matching set MP_a for the new image $I_{\boldsymbol{s}_a}$. Different ways to obtain the weights η_A and η_B may exist in order to satisfy the feature-position-consistent interpolation model in equation (4.5).

4.1.3 Tri-view feature matching

A large number of approximately uniformly distributed matchings are required for our affine transformation based view interpolation method. The Harris corners are

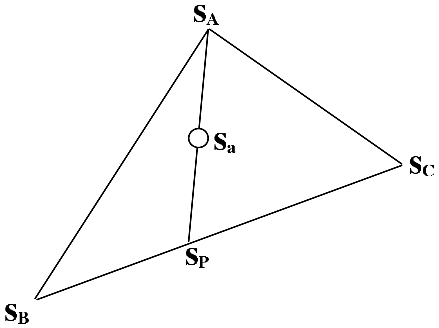


Figure 4.2: The geometric relationships between $\boldsymbol{s}_A$, $\boldsymbol{s}_B$, $\boldsymbol{s}_C$, $\boldsymbol{s}_P$ and $\boldsymbol{s}_a$.

first detected in the images $I_{\boldsymbol{s}_A}$, $I_{\boldsymbol{s}_B}$ and $I_{\boldsymbol{s}_C}$. Then, the two-view matchings between $I_{\boldsymbol{s}_A}$ and $I_{\boldsymbol{s}_B}$, $I_{\boldsymbol{s}_B}$ and $I_{\boldsymbol{s}_C}$, and $I_{\boldsymbol{s}_C}$ and $I_{\boldsymbol{s}_A}$ are found from the detected corners through normalized correlation and refined through the fundamental matrices. From the matchings between $I_{\boldsymbol{s}_A}$ and $I_{\boldsymbol{s}_B}$, and $I_{\boldsymbol{s}_B}$ and $I_{\boldsymbol{s}_C}$, we can set up the tri-view matching relationship among $I_{\boldsymbol{s}_A}$, $I_{\boldsymbol{s}_B}$ and $I_{\boldsymbol{s}_C}$, related through the common feature points in image $I_{\boldsymbol{s}_B}$. Finally, we use the matchings between $I_{\boldsymbol{s}_C}$ and $I_{\boldsymbol{s}_A}$ to check the validity of the above tri-view matchings. Experiments show that a set of good matchings can be obtained through the above methods. It is obvious that the proposed ABCA law can easily be extended to multi-view (more than three) matching detection.

In order to increase the number of matching feature points, we calculate the fundamental matrices between each two-view pair and the tri-view tensor from the above matchings. The matchings between each two-view pair will be checked with the correspondent new fundamental matrix and then more matchings can be obtained using tensor-based transferring from two-view matchings to the third view [15], [16]. In addition, the matchings that are inconsistent with their neighbors are removed.

We have obtained the feature matching relationship among the pre-captured views $I_{\boldsymbol{s}_A}$, $I_{\boldsymbol{s}_B}$, $I_{\boldsymbol{s}_C}$ and new view $I_{\boldsymbol{s}_a}$. Now we want to set up the relationship between triangular patches among these views.

The new view without texture is first partitioned using Delaunay triangulation [94] through the points $\boldsymbol{x}_{s_a}(\boldsymbol{x}_{s_A,n})$. The corresponding triangular patches in pre-captured image $\boldsymbol{I}_{\boldsymbol{s}_A}$, $\boldsymbol{I}_{\boldsymbol{s}_B}$ and $\boldsymbol{I}_{\boldsymbol{s}_C}$ can thus be obtained based on the Delaunay triangulation of the view $\boldsymbol{I}_{\boldsymbol{s}_a}$. The corresponding matchings (in $\boldsymbol{I}_{\boldsymbol{s}_A}$, $\boldsymbol{I}_{\boldsymbol{s}_B}$, $\boldsymbol{I}_{\boldsymbol{s}_C}$) of the three vertices of one triangular patch in $\boldsymbol{I}_{\boldsymbol{s}_a}$ construct the corresponding triangular patches in $\boldsymbol{I}_{\boldsymbol{s}_A}$, $\boldsymbol{I}_{\boldsymbol{s}_B}$, $\boldsymbol{I}_{\boldsymbol{s}_C}$. Obviously, the triangulations of pre-captured images generated in this way may not be exactly the Delaunay triangulations, but are approximate ones.

In this way, we set up the triangular patch relationships among the pre-captured images and the new view. Assume $\boldsymbol{T}^m_{s_a}(\boldsymbol{x}_{s_a,n_1},\ \boldsymbol{x}_{s_a,n_2}, \boldsymbol{x}_{s_a,n_3})$ denotes a triangular patch in image $\boldsymbol{I}_{\boldsymbol{s}_a}$ with three vertices $\boldsymbol{x}_{s_a,n_1}$, $\boldsymbol{x}_{s_a,n_2}$ and $\boldsymbol{x}_{s_a,n_3}$, $m = 1, 2, ..., M$ with M the total number of Delaunay triangles. The corresponding triangular patches in $\boldsymbol{I}_{\boldsymbol{s}_A}$ are $\boldsymbol{T}^m_{s_A}(\boldsymbol{x}_{s_A}(\boldsymbol{x}_{s_a,n_1}),\ \boldsymbol{x}_{s_A}(\boldsymbol{x}_{s_a,n_2}),\ \boldsymbol{x}_{s_A}(\boldsymbol{x}_{s_a,n_3}))$, and similarly $\boldsymbol{T}^m_{s_B}$, $\boldsymbol{T}^m_{s_C}$ for images $\boldsymbol{I}_{\boldsymbol{s}_B}$ and $\boldsymbol{I}_{\boldsymbol{s}_C}$. From now on, we will use $\boldsymbol{T}^m_{s_a}$, $\boldsymbol{T}^m_{s_A}$, $\boldsymbol{T}^m_{s_B}$ and $\boldsymbol{T}^m_{s_C}$ to denote a set of corresponding triangular patches.

4.1.4 Texture rendering methods for different categories of triangular patches

The affine transformation is used for texture mapping. The affine transformation for texture mapping is a good model under the following conditions: 1) the triangular patch is physically located in a plane in the scene; or 2) the triangular patch is small enough; or 3) the separations between the camera positions where pre-captured images are taken are small enough. Then the corresponding texture of the triangular patch in the new view can be mapped from the three pre-captured images. The affine transformation can be determined from the geometric relationship between the positions of the three corresponding vertices.

A six parameter affine transformation $\boldsymbol{t}^A_m$ can be obtained from the geometric relationship between three corresponding vertices of triangular patches $\boldsymbol{T}^m_{s_a}$ and $\boldsymbol{T}^m_{s_A}$, i.e., between $(\boldsymbol{x}_{s_a,n_1},\ \boldsymbol{x}_{s_a,n_2}, \boldsymbol{x}_{s_a,n_3})$ and $(\boldsymbol{x}_{s_A}(\boldsymbol{x}_{s_a,n_1}), \boldsymbol{x}_{s_A}(\boldsymbol{x}_{s_a,n_2}),\ \boldsymbol{x}_{s_A}(\boldsymbol{x}_{s_a,n_3}))$. The relationship can be represented by

$$
\begin{aligned}
\boldsymbol{x}_{s_a,n_1} &= \begin{bmatrix} t^A_{m,1} & t^A_{m,2} \\ t^A_{m,3} & t^A_{m,4} \end{bmatrix} \boldsymbol{x}_{s_A}(\boldsymbol{x}_{s_a,n_1}) + \begin{bmatrix} t^A_{m,5} \\ t^A_{m,6} \end{bmatrix} \\
\boldsymbol{x}_{s_a,n_2} &= \begin{bmatrix} t^A_{m,1} & t^A_{m,2} \\ t^A_{m,3} & t^A_{m,4} \end{bmatrix} \boldsymbol{x}_{s_A}(\boldsymbol{x}_{s_a,n_2}) + \begin{bmatrix} t^A_{m,5} \\ t^A_{m,6} \end{bmatrix} \\
\boldsymbol{x}_{s_a,n_3} &= \begin{bmatrix} t^A_{m,1} & t^A_{m,2} \\ t^A_{m,3} & t^A_{m,4} \end{bmatrix} \boldsymbol{x}_{s_A}(\boldsymbol{x}_{s_a,n_3}) + \begin{bmatrix} t^A_{m,5} \\ t^A_{m,6} \end{bmatrix}
\end{aligned} \tag{4.8}
$$

where $\boldsymbol{t}^A_m = [t^A_{m,1}\ t^A_{m,2}\ t^A_{m,3}\ t^A_{m,4}\ t^A_{m,5}\ t^A_{m,6}]^T$. There are 6 unknown variables in the above 6 equations, thus we can obtain $\boldsymbol{t}^A_m$ by solving these 6 linear equations for a particular triangular patch $\boldsymbol{T}^m_{s_a}$.

If we use $E_{s_a}(\boldsymbol{x})$ $(\boldsymbol{x} \in \boldsymbol{T}^m_{s_a})$ to denote the texture within the triangular patch $\boldsymbol{T}^m_{s_a}$, then $E_{s_A}(\boldsymbol{x}) = F(\boldsymbol{x}, I_{\boldsymbol{s}_A}, \boldsymbol{t}^A_m)$ $(\boldsymbol{x} \in \boldsymbol{T}^m_{s_a})$ represents obtaining the texture of $\boldsymbol{T}^m_{s_a}$ from $I_{\boldsymbol{s}_A}$ through affine transformation $\boldsymbol{t}^A_m$.

$$
E_{s_A}(\boldsymbol{x}) = (\mathcal{H} I_{\boldsymbol{s}_A})(\begin{bmatrix} t^A_{m,1} & t^A_{m,2} \\ t^A_{m,3} & t^A_{m,4} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} t^A_{m,5} \\ t^A_{m,6} \end{bmatrix}), \boldsymbol{x} \in \boldsymbol{T}^m_{s_a} \tag{4.9}
$$

where $\mathcal{H}$ is a linear interpolation operator.

Similarly, we can have $E_{s_B}(\boldsymbol{x}) = F(\boldsymbol{x}, I_{\boldsymbol{s}_B}, \boldsymbol{t}^B_m)$ and $E_{s_C}(\boldsymbol{x}) = F(\boldsymbol{x}, I_{\boldsymbol{s}_C}, \boldsymbol{t}^C_m)$, $\boldsymbol{x} \in \boldsymbol{T}^m_{s_a}$. The desired texture $E(\boldsymbol{x})$, $\boldsymbol{x} \in \boldsymbol{T}^m_{s_a}$, can be obtained from $E_{s_A}(\boldsymbol{x})$, $E_{s_B}(\boldsymbol{x})$, $E_{s_C}(\boldsymbol{x})$ or a combination of them. One simple way to generate the texture in triangular patch $\boldsymbol{T}^m_{s_a}$ is

$$
E(\boldsymbol{x}) = \alpha_A \cdot E_{s_A}(\boldsymbol{x}) + \alpha_B \cdot E_{s_B}(\boldsymbol{x}) + \alpha_C \cdot E_{s_C}(\boldsymbol{x}) \tag{4.10}
$$

for any $\boldsymbol{x} \in \boldsymbol{T}^m_{s_a}$. $\alpha_A = 1/(1+\eta_B)$, $\alpha_B = \eta_B/(1+\eta_A)(1+\eta_B)$ and $\alpha_C = \eta_A \cdot \eta_B/(1+\eta_A)(1+\eta_B)$. η_A and η_B are calculated based on equation (4.7).

However, the texture differences between $E_{s_A}(\boldsymbol{x})$, $E_{s_B}(\boldsymbol{x})$ and $E_{s_c}(\boldsymbol{x})$ in equation (4.10) are potentially significant for some triangular patches. This may bring ghosting artifacts [43] in the synthesized images. In order to avoid such side effect and thus to minimize the discontinuities between the triangular patches in the new view, the following rendering strategy is used,

$$E(\boldsymbol{x}) = \begin{cases} E_{s_A}(\boldsymbol{x}) & \text{if } d_A \leq \min(d_B, d_C) \\ E_{s_B}(\boldsymbol{x}) & \text{else if } d_B < \min(d_A, d_C) \\ E_{s_C}(\boldsymbol{x}) & \text{otherwise} \end{cases}$$

for any $\boldsymbol{x} \in \boldsymbol{T}^m_{s_a}$. $d_A = (\sum_{n=1}^{N} |\boldsymbol{x}_{s_a}(\boldsymbol{x}_{s_A,n}) - \boldsymbol{x}_{s_A,n}|)$, $d_B = (\sum_{n=1}^{N} |\boldsymbol{x}_{s_a}(\boldsymbol{x}_{s_A,n}) - \boldsymbol{x}_{s_B}(\boldsymbol{x}_{s_A,n})|)$, and $d_C = (\sum_{n=1}^{N} |\boldsymbol{x}_{s_a}(\boldsymbol{x}_{s_A,n}) - \boldsymbol{x}_{s_C}(\boldsymbol{x}_{s_A,n})|)$. Similar idea for texture mapping appears in [95], [96], [13].

4.1.5 Simulation results

Simulations of the proposed view interpolation algorithm include three parts. The first part is algorithm validation. In this part, we apply the proposed method on the situation with only two pre-captured images. A view in-between these two views is known and represents ground-truth. The in-between view is first interpolated by the proposed algorithm and then compared to the ground-truth. The second part is algorithm illustration. Three pre-captured images are used to illustrate the procedure as to how the interpolated view can be obtained. In the last part, or algorithm application part, the proposed algorithm is applied on three pre-captured images that were taken in a real environment.

Due to the nature of IBR applications, physically valid views are not necessarily required during navigation, if obtaining such views is impossible or too expensive. When a user navigates in a real-image-based virtual environment, the corresponding view changing during navigation is the desired experience that he (she) expects. Thus, the two important requirements are : (1) the views are of good quality, and (2) view changing during navigation is done in a natural fashion. If these two requirements are met, it is likely to be acceptable for most IBR applications.

For traditional image and video processing applications, the evaluation methods include both **subjective evaluation** methods and **objective evaluation** methods.

One of the widely-used objective evaluation methods is based on PSNR (peak signal to noise ratio) if the ground-truth images for comparison are known. However, the PSNR may not be an appropriate metric to evaluate the synthesized views for IBR applications even if the ground-truth views are known. The higher PSNR values of the synthesized views (comparing to the ground-truth views) may not necessarily mean that the synthesized views are of better subjective quality. In addition, it is very difficult or expensive to obtain the ground-truth views for comparison in the scenario of view interpolation that we discussed here. Thus, we will use informal subjective evaluation to evaluate our simulation results.

Algorithm validation

View interpolation based on two source images has been studied for a long time. The methods in the literature are usually based on dense disparity maps between the two source images. For algorithm validation purpose, we implement the proposed algorithm on this situation to show that the proposed method can generate valid interpolated views.

We choose images No. 21 and No. 23 in the flower garden sequence as the source images I_1 and I_2 (these images are individual frames which have been reconstructed from the original interlaced sequence and downsampled). The source images are shown in Fig. 4.3.

We can interpolate the images between I_1 and I_2 based on the proposed algorithm. For this comparison, we interpolate the middle view corresponding to image No. 22 in the sequence. The interpolated view is shown at the top of Fig. 4.4. For visual comparison, the original intermediate image, image No.22 in the garden flower sequence, is given at the bottom of Fig. 4.4. From the above simulation results, we can see that the proposed algorithm can generate a valid interpolated intermediate view.

Due to the fact that the matching features may not be found at the border areas in the original images (i.e., I_1 and I_2), the texture at the border areas in the synthesized

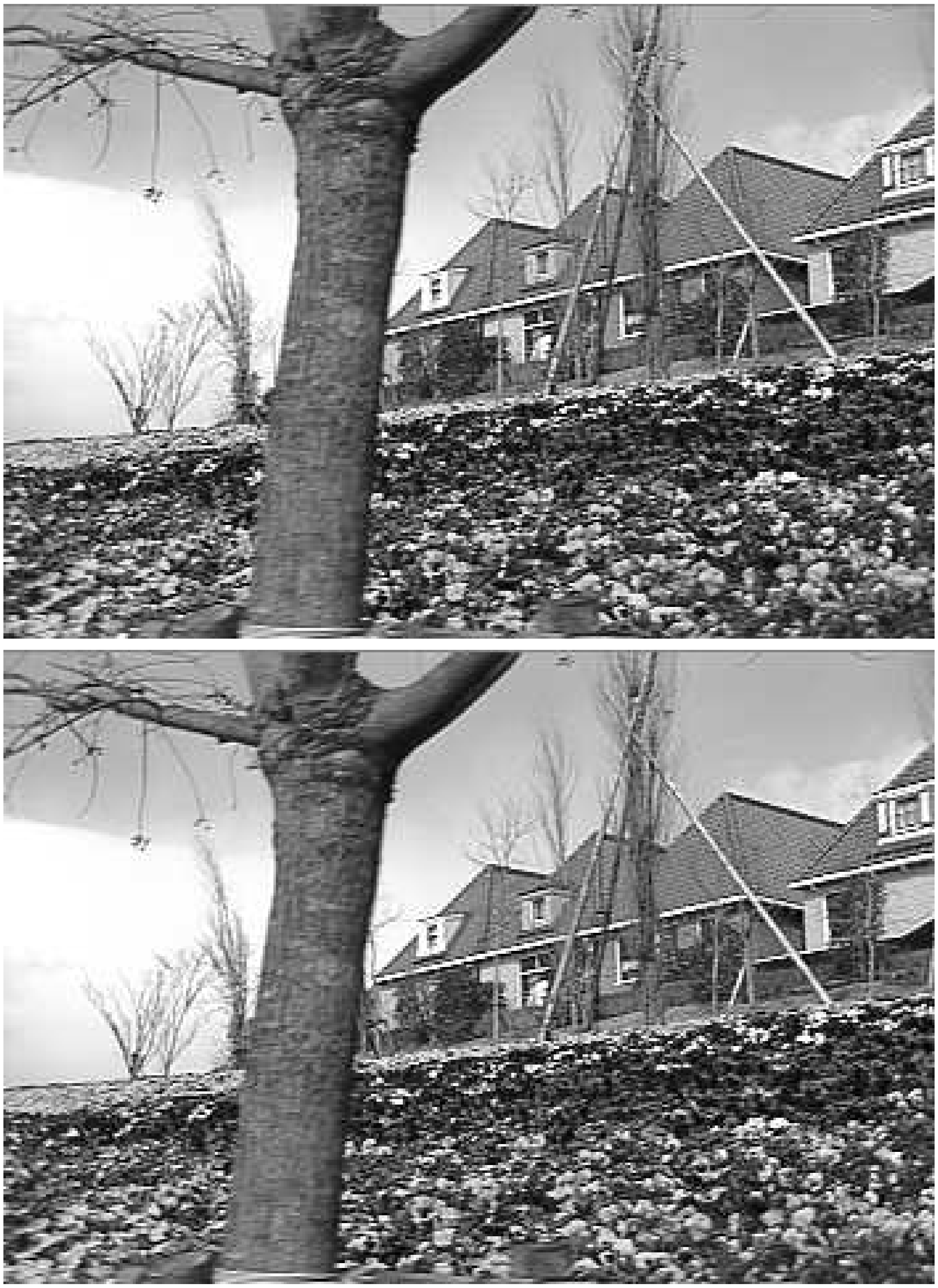


Figure 4.3: Source image I_1 (flower garden 21) and image I_2(flower garden 23)

views is missing. This issue will be addressed later. The image looks blurred near the edge of trunk area (the area within the circle shown in the figure). This is caused by occlusions between I_1 and I_2. The occlusion detection is not a research topic in this thesis. The image is generally of good quality in the other areas.

In fact, the proposed algorithm is equivalent to the dense-disparity-based view interpolation method for view interpolation with two source views. The dense disparity is obtained by piecewise planar interpolation of the feature-point disparities. For the scenarios that there are more than two source images, the dense disparity maps are usually very difficult to obtain but the proposed algorithm can avoid these difficulties due to the use of the matching features.

Algorithm Illustration

In the second simulation, the procedure of the proposed view interpolation algorithm is illustrated on a scene of a model house on a table (obtained from [97]), shown in Fig. 4.5, where different views of the model house were captured. In Fig. 4.6, the original images (pictures that were taken at different viewing positions and from different viewing directions) are shown in Fig. 4.6(a), Fig. 4.6(b) and Fig. 4.6(c). We have no information regarding to the relative positions where the image sequence are taken. From the view changing, we can reasonably estimate that the camera was moving on a trajectory which is similar to that shown in Fig. 4.5. The arrows show the imaging directions at different camera positions. The matching features among three original images are found and the corresponding features located in the view to be interpolated (the position $\boldsymbol{s}_a$ is selected as the center of the triangle $\boldsymbol{s}_A\boldsymbol{s}_B\boldsymbol{s}_C$) can be computed, as shown in Fig. 4.6-(d).

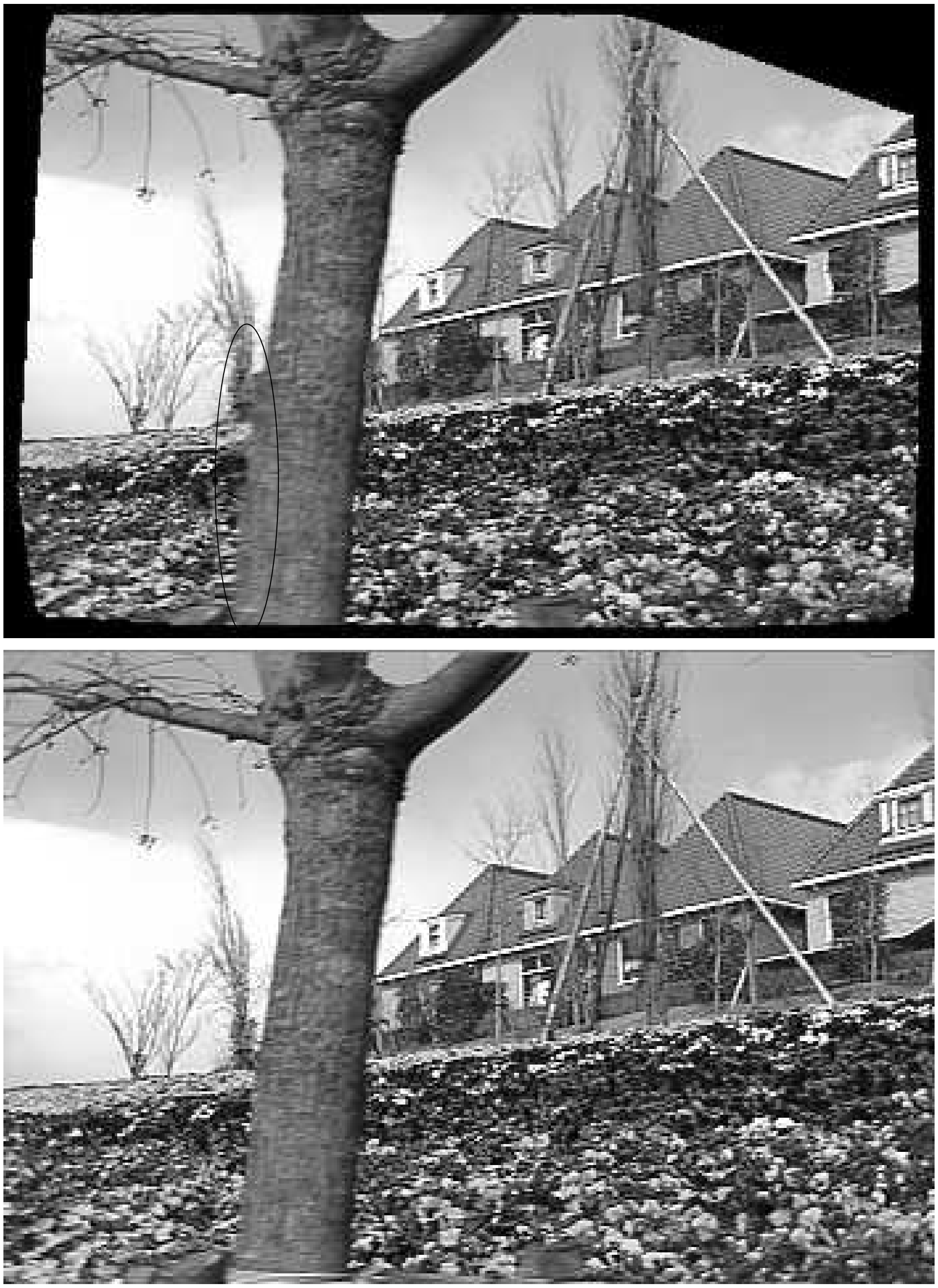


Figure 4.4: The view interpolated from I_1 and I_2 using the proposed algorithm (top); the original intermediate image(bottom)

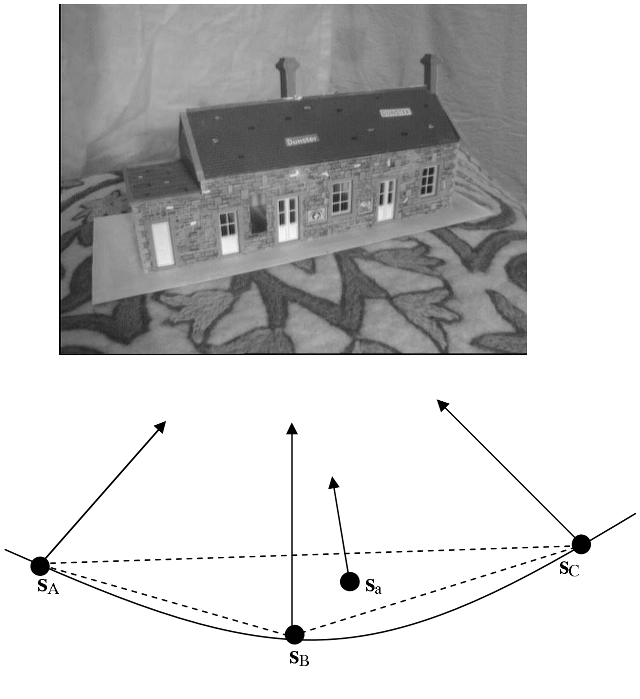
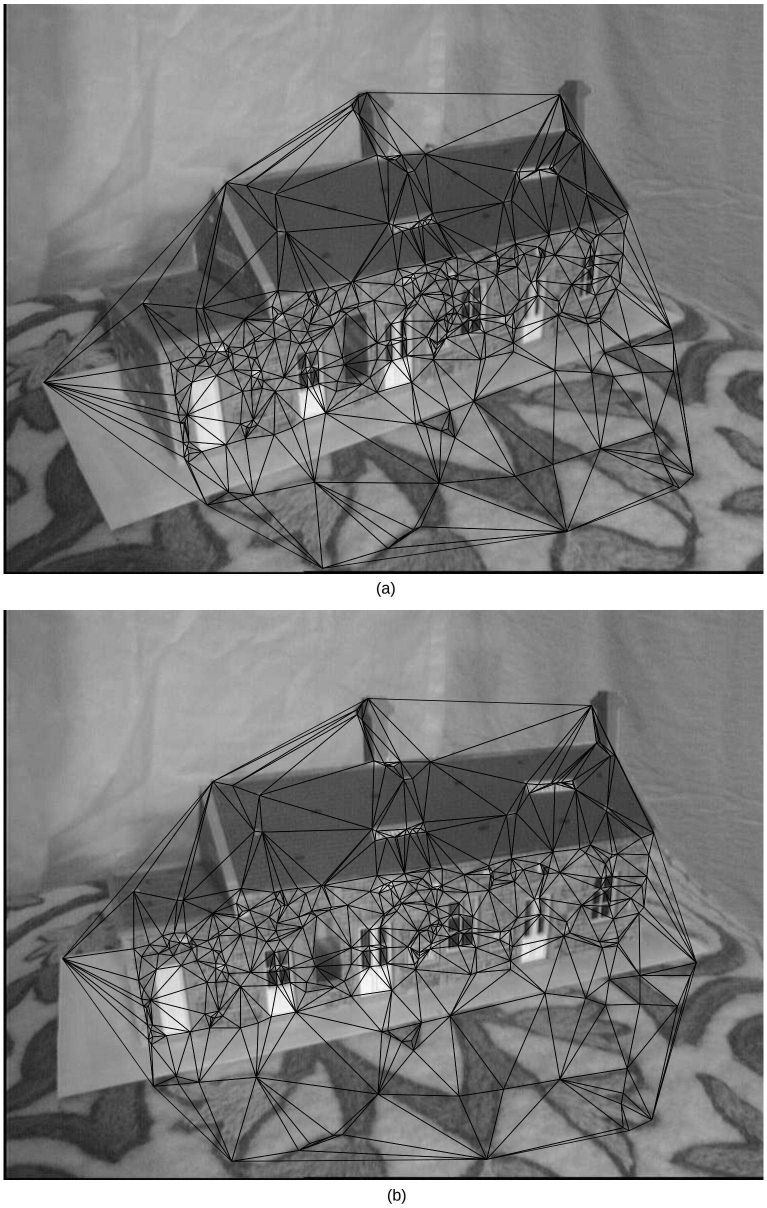


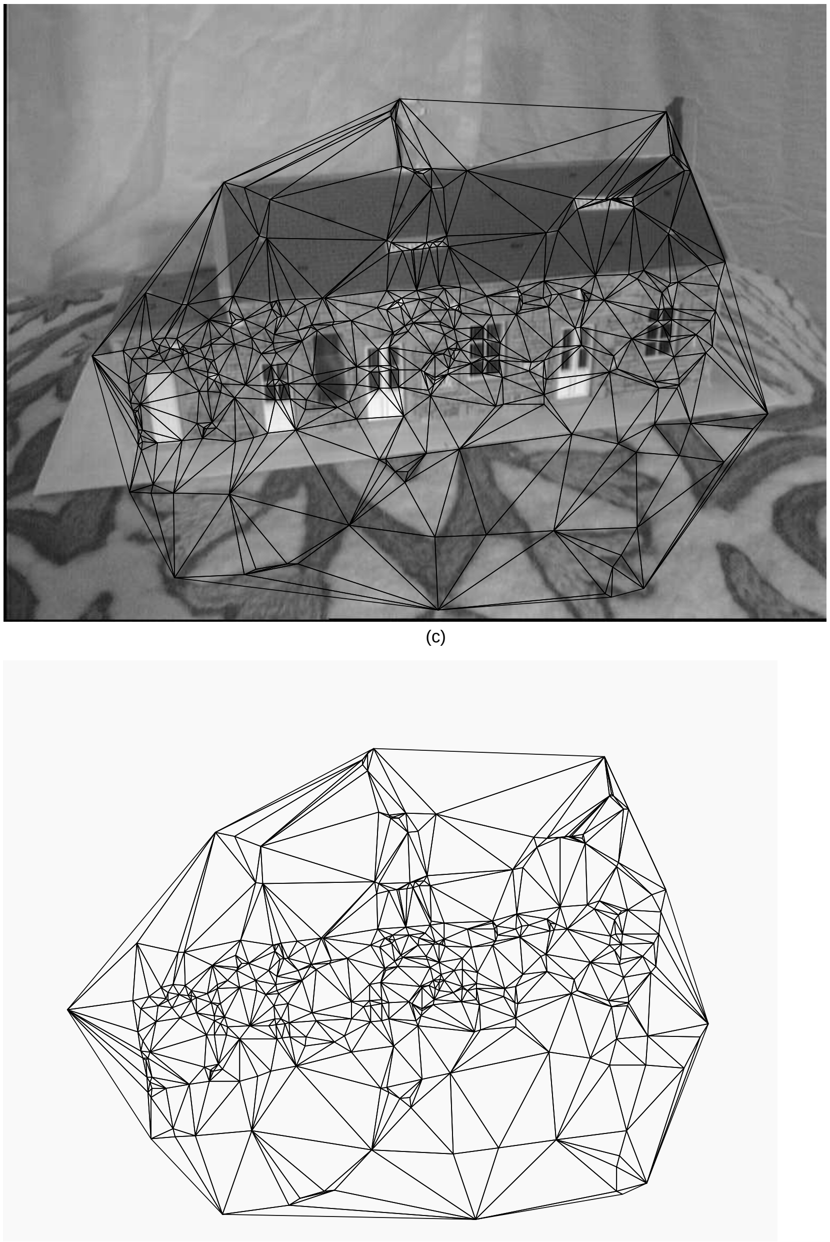
Figure 4.5: Illustration of the estimated camera trajectory.



(a)

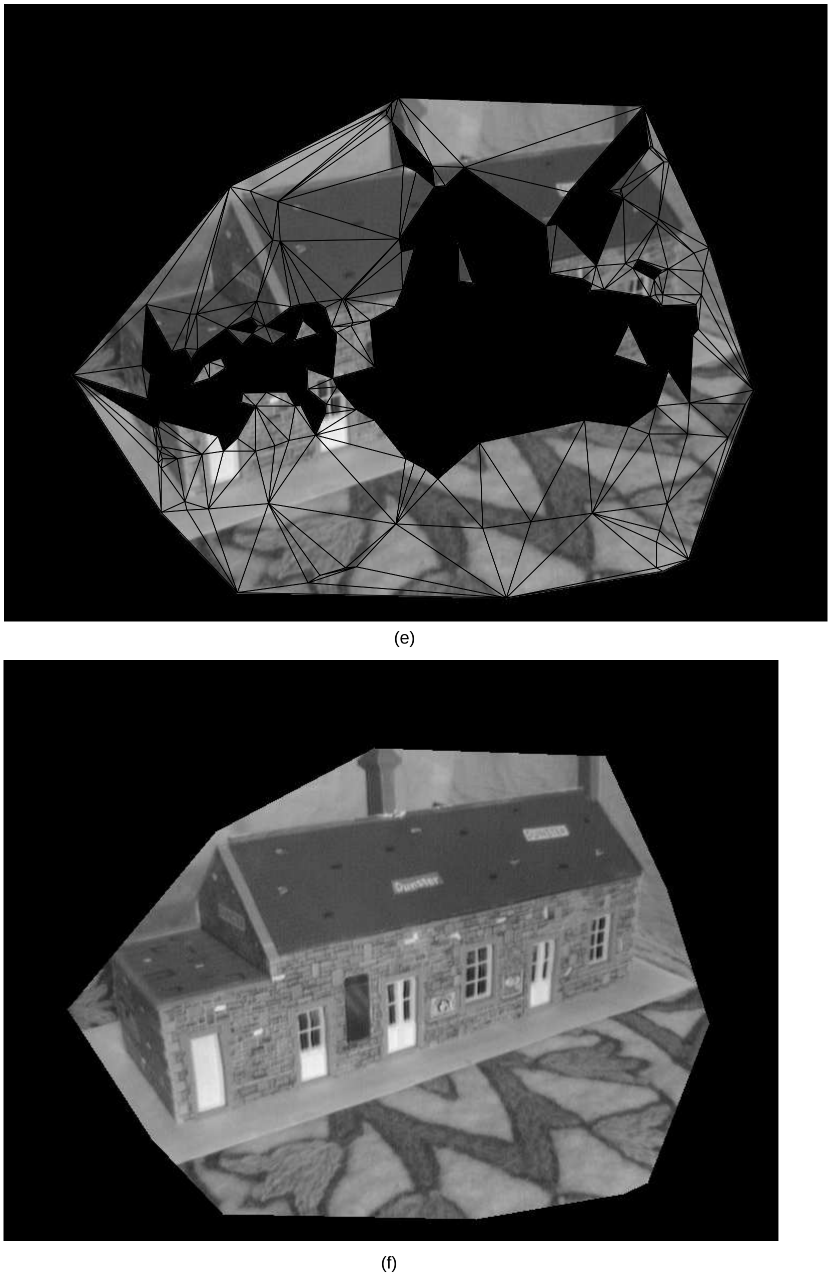
(b)

Figure 4.6: Illustration of the proposed view-interpolation algorithm. (a) The first original image (with triangular mesh). (b) The second original image (with triangular mesh).



(c)

Figure 4.6 Illustration of the proposed view-interpolation algorithm. (c) The third original image (with triangular mesh). (d) The triangular mesh on the view to be interpolated (no texture).



(e)

(f)

Figure 4.6 Illustration of the proposed view-interpolation algorithm. (e) Filling texture within each triangular area one by one. (f) The final interpolated view (after all triangular areas have been filled with texture from original view).

The view to be interpolated is triangulated with the above matching features, as shown in Fig. 4.6-(d), but without any texture. The original images in Fig. 4.6-(a), Fig. 4.6-(b) and Fig. 4.6-(c) are segmented into the triangular areas, respectively, in the same way as that of the view to be interpolated. The triangulation results are superimposed on the images in Fig. 4.6-(a), Fig. 4.6-(b) and Fig. 4.6-(c). In this way, the relationship between corresponding region in different views is built and thus the texture mapping can be implemented.

Fig. 4.6-(e) shows the view to be interpolated, which is only partially filled up with texture. The final interpolated view is shown in Fig. 4.6-(f). We can see that the synthesized image is a valid view of good quality except the missing texture at the border areas.

Algorithm applications

The third simulation was implemented on three pre-captured images of size 1024×768, taken at three positions in the VIVA lab at the University of Ottawa. The relative camera positions are shown in Fig. 4.7 with $s_A s_B \simeq s_A s_C \simeq s_B s_C \simeq 50(cm)$. The three pre-captured images (with similar imaging directions) are shown in Fig. 4.8(a), Fig. 4.8(b) and Fig. 4.8(c). 3595 matching features were detected and the images were segmented into 7893 triangular patches. Thus the area of each triangular patch is relatively small, which is suitable for the affine transformation model. View interpolation using the proposed algorithm is thus implemented. One of the interpolated views is shown in Fig. 4.8(d). From the figure, we can see that the synthesized view is of good quality and can represent the corresponding change of view. Thus, it satisfies our objective. Discontinuities appear around some areas (highlighted by the circles in the figure) in the synthesized view such as the table border area (central bottom area in the image). This is caused by potential faulty matching points near the image border.

We note that there is missing texture in the border part of the interpolated views. This is due to the lack of matching features in these areas. It will be not an issue if

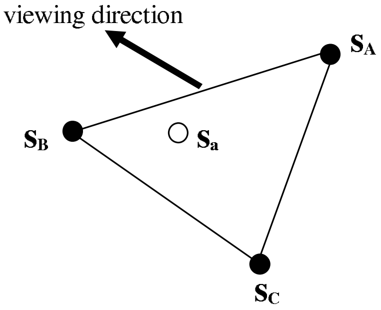
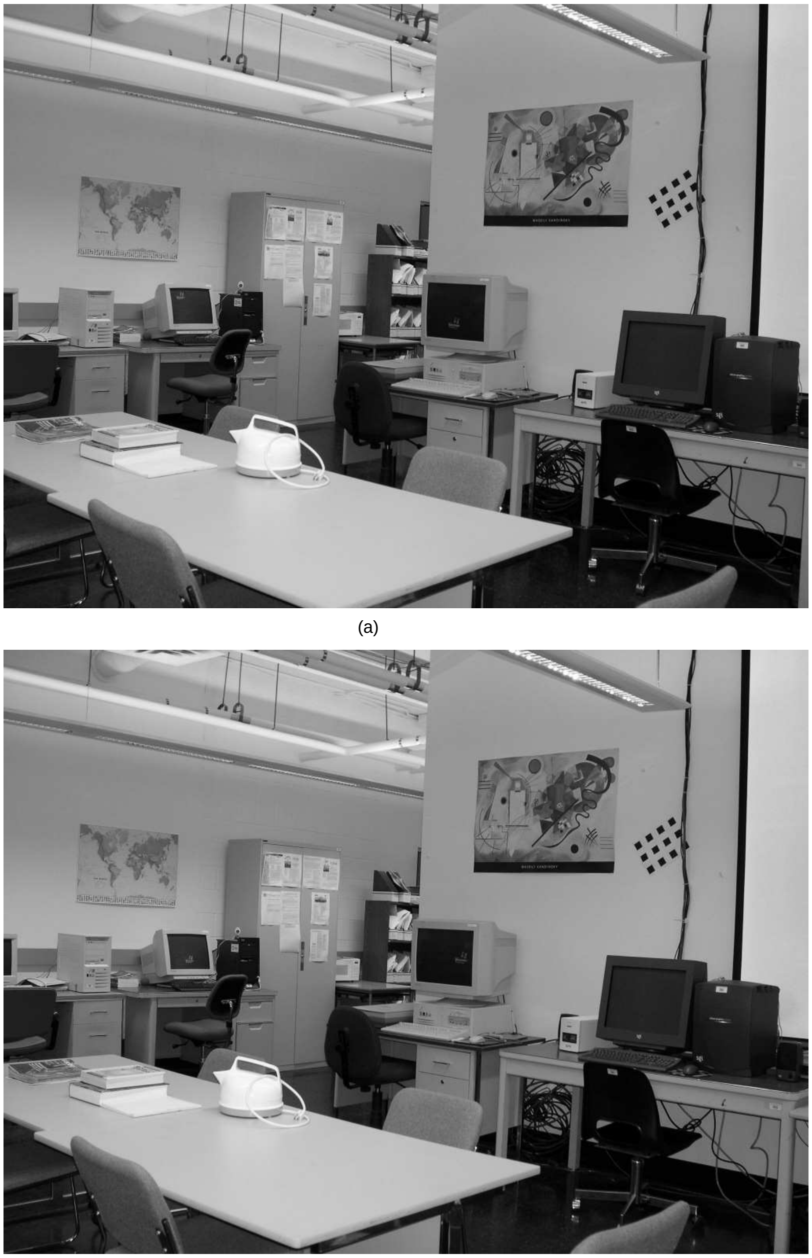


Figure 4.7: The illustration of the camera positions where the pre-captured images were taken ($\boldsymbol{s}_A$, $\boldsymbol{s}_B$ and $\boldsymbol{s}_C$)and the virtual camera position where the synthesized view is captured $\boldsymbol{s}_a$.

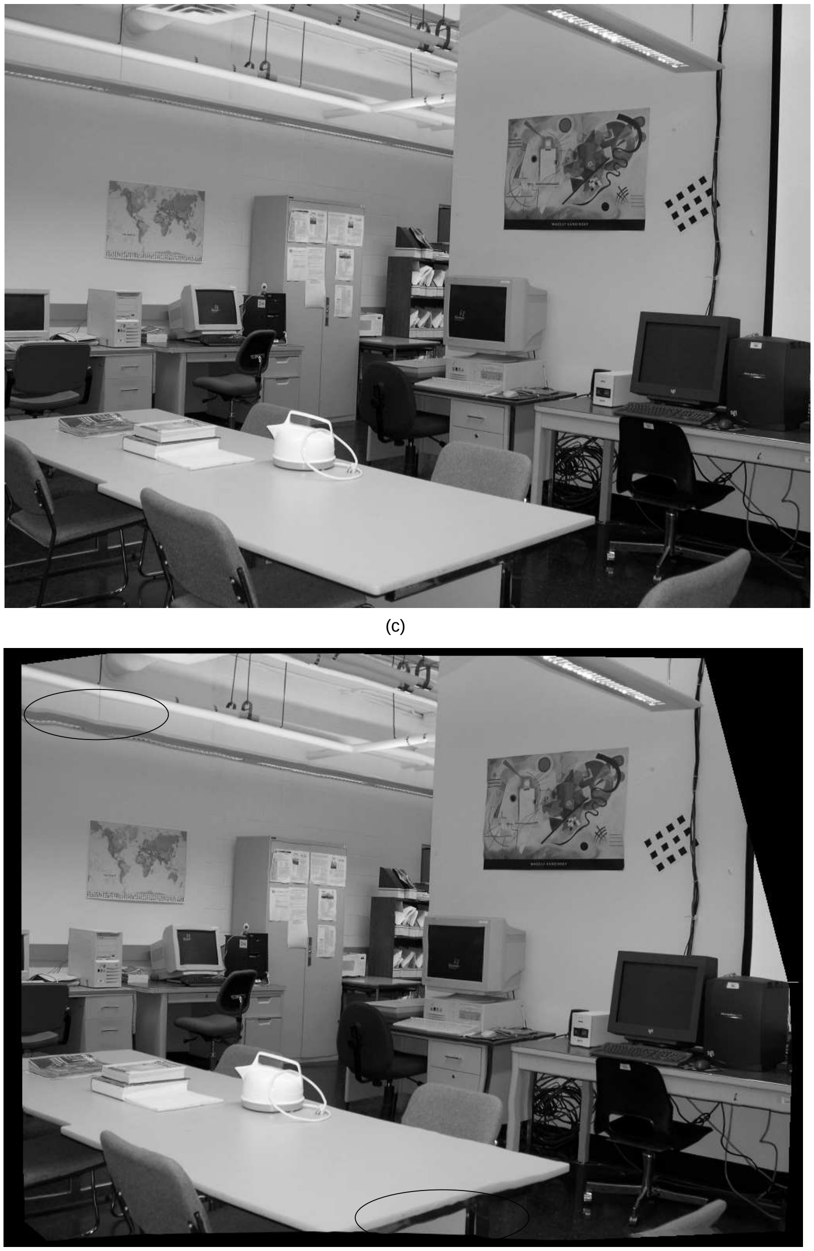
the source images are large enough, and in particular if panoramas are used as source images in IBR applications.



(a)

(b)

Figure 4.8: The simulation result in a real environment using the proposed view-interpolation algorithm. (a) The first original image. (b) The second original image.



(c)

Figure 4.8 The simulation result in a real environment using the proposed view-interpolation algorithm. (c) The third original image. (d) One of the synthesized views using the proposed method.

4.2 Scenario II: A new dense disparity estimation model for IBR applications

Disparity estimation between multiple views is a fundamental problem that has been studied for applications in 3D reconstruction [98], [99], view interpolation [100], multiview coding [101], etc. Much research has been carried out and different methods for optical flow estimation or dense disparity estimation have been proposed over many years. In most methods, the disparities between multiple views are assumed to be small in order to use a gradient-based approach. In many special applications, two source views are even assumed to be taken under a parallel-camera-configuration and thus dense matching sometimes refers to stereo matching [23].

View interpolation under more general camera movement situations is desired for IBR applications. Often, the small-disparity assumption may not be satisfied. Although methods have been proposed to obtain an initial coarse estimation before applying the above gradient-based approaches, these algorithms can fail in some situations in IBR applications. In this section, a new dense disparity estimation model is proposed in order to use the traditional methods to solve dense disparity estimation problems in such new situations. In the proposed method, the dense disparity estimation is implemented using transformed views which are obtained through minimizing the difference between the source views. The dense disparity maps between two original source images are computed afterward from an inverse transformation from the above estimates. In this thesis, we use transformed views to estimate the disparities instead of using two source views based on an initial estimation because experiments show that the large difference of sampling densities between two source views makes the matching unreliable.

4.2.1 Review of previous major dense disparity estimation methods and their limitations

Assume that we want to estimate the disparities between two discrete images (for example, the left and right images in a stereo pair) $I_l(\boldsymbol{m})$ and $I_r(\boldsymbol{m})$. The disparities $\{\boldsymbol{d}(\boldsymbol{m})\}_{\boldsymbol{m}\in W\cap\Lambda}$ are defined on a lattice Λ in the common area W of image $I_l(\boldsymbol{m})$ and $I_r(\boldsymbol{m})$. The disparities between these two images are usually estimated by minimizing a cost function,

$$E_D(\boldsymbol{d}) = \sum_{\boldsymbol{m}\in\mathcal{W}\cap\Lambda} E_D^{\boldsymbol{m}}(\boldsymbol{d}) \tag{4.11}$$

where

$$E_D^{\boldsymbol{m}}(\boldsymbol{d}) = \sum_{\boldsymbol{n}\in\mathcal{C}_{\boldsymbol{m}}} \Psi(I_l(\boldsymbol{n}) - (\mathcal{H}I_r)(\boldsymbol{n}+\boldsymbol{d}(\boldsymbol{m}))) \tag{4.12}$$

$\Psi(\cdot)$ represents a metric function, such as $|\cdot|$ or $(\cdot)^2$. We will use $(\cdot)^2$ here but the method is not limited to this metric function. $\mathcal{H}$ represents a linear interpolation operator to obtain a continuous version of image I_r since in general $\boldsymbol{n}+\boldsymbol{d}(\boldsymbol{m})$ does not lie on the sampling lattice Λ. $\mathcal{C}_{\boldsymbol{m}} \subset \Lambda$ is a neighbor area around $\boldsymbol{m}$. In some algorithms, a filter or filter banks are used to process the original images in order to make better use of the neighborhood feature information. In this thesis, the proposed algorithm will be illustrated using the basic format in Eq. (4.12).

Eq. (4.12) provides the data fidelity term in disparity estimation. In order to obtain a reliable result, some constraints have to be added as a regularization term, denoted by $E_S(\boldsymbol{d})$. Thus, the cost function for optimization is $E(\boldsymbol{d}) = E_D(\boldsymbol{d}) + \lambda E_S(\boldsymbol{d})$, where λ is a regularization coefficient. There are many ways to select the regularization term $E_S(\boldsymbol{d})$ proposed in the literature. A simple and commonly used constraint on $\bigtriangledown\boldsymbol{d}$ is chosen without loss of generality in this thesis.

One major method [102] to minimize Eq. (4.12) is expanding $(\mathcal{H}I_r)(\boldsymbol{n}+\boldsymbol{d}(\boldsymbol{m}))$ around $\boldsymbol{n}$ using methods such as Taylor series expansion [103]. Other methods such as using the gradient-descent-based algorithm to solve the associated Euler-Lagrange equation [100] are also widely studied. All these methods must start with some initial disparity estimates when the disparities between views are large. The estimation is then implemented in a coarse-refine routine. Obtaining a good coarse estimation of the disparity field is very difficult and expensive when the disparities between views are large. The situation is even worse when the disparity variations are large. On the other hand, the initial disparity estimates may be obtained considering the physical imaging conditions.

The previous algorithms in the literature do not work well in some situations, such as when two views are taken by a camera moving forward or backward with a large relative distance. In such situations, matching of the areas with different sampling densities in the source views of the similar scenes is not reliable even though a good estimation is available.

Based on the above observation, a new disparity estimation method is proposed. The sampling-density difference of two source views is first minimized through a suitable transformation which is determined by the physical imaging conditions. The proposed method is currently focused on view interpolation between two views taken by a camera moving forward or backward.

4.2.2 The proposed new dense disparity estimation model

From the above analysis, we find that a general pre-processing to reduce the difference between I_l and I_r will be very helpful. A new image is defined through changing the sampling structure of image I_l as $\widetilde{I}_l(\boldsymbol{m}) = (\mathcal{H}I_l)(\mathcal{M}\boldsymbol{m})$. $\mathcal{M}$ is a general transformation which is selected to reduce the disparities between $\widetilde{I}_l$ and I_r. Then, the dense disparities can be calculated between $\widetilde{I}_l$ and I_r using the previous methods such as the algorithms in [103], [100] (based on equation (4.11) and (4.12)),

$$E_D^{\boldsymbol{m}}(\widetilde{\boldsymbol{d}}) = \sum_{\boldsymbol{n}\in\mathcal{C}_{\boldsymbol{m}}} (\widetilde{I}_l(\boldsymbol{n}) - (\mathcal{H}I_r)(\boldsymbol{n}+\widetilde{\boldsymbol{d}}(\boldsymbol{m})))^2 \tag{4.13}$$

Assume that the optimal disparity field between $\widetilde{I}_l$ and I_r is $\{\hat{\widetilde{\boldsymbol{d}}}\}$, or $\widetilde{I}_l(\boldsymbol{m}) \cong (\mathcal{H}I_r)(\boldsymbol{m}+\hat{\widetilde{\boldsymbol{d}}}(\boldsymbol{m}))$. Then, the disparities $\boldsymbol{d}$ between I_l and I_r can be obtained as

$$\boldsymbol{d} = \mathcal{M}^{-1}\boldsymbol{m} - \boldsymbol{m} + (\mathcal{H}\hat{\widetilde{\boldsymbol{d}}})(\mathcal{M}^{-1}\boldsymbol{m}) \tag{4.14}$$

where $\mathcal{M}^{-1}$ denotes the inverse transformation of $\mathcal{M}$. Similarly, because the disparity $\hat{\widetilde{\boldsymbol{d}}}$ is defined on a lattice, we need a linear interpolator to get its values at the positions off the lattice.

Some previous work in the literature can be essentially included in the above preprocessing strategy. For example, view rectification is applied for view interpolation through view warping [35]. After rectification, two rectified views can be regarded as being taken by a pair of parallel camera. Then the disparity estimation can be converted from 2D to 1D.

4.2.3 Two Major transformations for view interpolation in IBR applications

Two major types of camera movement, i.e., translation perpendicular to the camera axis and forward or backward motion, are of interest for view interpolation in IBR. The previous algorithms in the literature (such as [103], [100]) can usually obtain good results if the camera movement is translation. We reformulate these previous methods into our model and use a region-based approach, which is more stable compared to pixel-based or block-based (assuming the block size is much smaller than the region) approaches. The significant contribution of the above disparity-estimation model is for the situation of forward or backward camera movement, which may make previous algorithms fail. In such situations, the proper transformation, or zooming (up-sampling or down-sampling), is selected considering the physical imaging conditions. In addition, the difference of the two source views' sampling densities is minimized by warping one view, where the disparity estimation is actually carried out based on such a warped view. More general transformations, such as affine transformation, can also be used in some other imaging conditions.

Region-based dense matching

Block-based motion estimation algorithms [104], [105], [106] have been widely used in video compression applications [107]. For these applications, the sizes of blocks are usually small (such as 16 × 16, 8 × 8, 4 × 4). The estimated motion vector on a particular block is used to represent the motion of this whole block. In the proposed region-based dense matching algorithm, the sizes of the regions are large than the sizes of the above blocks. The dense matching is carried out between corresponding regions in the different images.

In region-based dense matching, one of the two source images (i.e., left image I_l) is divided into overlapped regions. These overlapped regions can be of arbitrary shape. They are partial images (sub-images). To simplify the algorithm, we use rectangular area to select these overlapped regions. In this way, the overlapped regions can be regarded as overlapped blocks with large size. One example of how to select these overlapped regions will be provided in the following simulations. The corresponding regions in the other image (i.e., right image I_r) can be found through a correlation-based search. Then, the disparities can be estimated on each pair of sub-images (or regions) and the disparities between two original images I_l and I_r can be recovered. The approach is very similar to the block-based approach. However, since the sizes of regions are very large, the disparities between each region pair are stable and thus the disparities for each pixel between each region pair are consistent with those between the original images. That is the reason we name our approach as region-based approach instead of block-based approach.

For the i-th region pair $I_{l,i}$ and $I_{r,i}$, assume that the approximate translation between this region pair is $\boldsymbol{d}_{0,i}$. Then the disparity estimation is carried out between $\widetilde{I}_{l,i}(\boldsymbol{m}) = (\mathcal{H}I_{l,i})(\boldsymbol{m} + \boldsymbol{d}_{0,i})$ and $I_{r,i}(\boldsymbol{m})$. The transformation $\mathcal{M}$ is defined as $\mathcal{M}\boldsymbol{m} = R_0 \cdot \boldsymbol{m} + \boldsymbol{d}_{0,i}$, where R_0 is identity matrix. The disparities between original images at corresponding pixels can be calculated through an inverse transformation. Because the regions are overlapped, the boundary problem (some scene areas may appear in one sub-image but do not appear in the other if the region sizes are equal) can be avoided by discarding the estimates in the boundary areas in each region pair.

Zooming-based transformation

The disparities between two views taken by a camera moving forward or backward have some special distributions. Even if there is no depth variation in the scene, the disparities are not uniform. The central part has very small disparities and the boundary part has large disparities; the situation is exactly the same as the disparities between an image and its zoomed version. Moreover, the differences of sampling densities at the boundary areas of such two images are usually very large, which makes matching very difficult due to interpolation $\mathcal{H}$. The situation becomes more complex with depth variations of the scene. However, a zooming function (up-sampling or down-sampling) usually provides a good transformation model for this kind of camera movement and the proposed disparity estimation model can avoid such difficulties.

Assume two images I_f (front image) and I_b (back image) are taken by a camera that moves backward (the center of the image is along the camera axis of zoom). An optimal zooming factor $\hat{\eta}$ is first found through

$$\hat{\eta} = \arg\min_{\eta} |I_b(\boldsymbol{m}) - (\mathcal{H}I_f)(\eta\boldsymbol{m})|^2. \tag{4.15}$$

The correspondent transformation is $\mathcal{M}\boldsymbol{m} = \begin{bmatrix} \hat{\eta} & 0 \\ 0 & \hat{\eta} \end{bmatrix} \cdot \boldsymbol{m} = \hat{\eta}\boldsymbol{m}$. The disparity field between $I_b(\boldsymbol{m})$ and $(\mathcal{H}I_f)(\hat{\eta}\boldsymbol{m})$ can be estimated using the previous methods [100], [103]. The disparity field between two original images I_f and I_b can then be calculated through the correspondent inverse transformation.

4.2.4 Simulation results

The simulations were implemented on a pair of images that were taken by a camera which moved forward and translated along both horizontal and vertical directions. The camera movement was manually controlled, so that the imaging planes of the camera at two different positions are roughly parallel. The two source images I_f and I_b are shown in Fig. 4.9 in reduced size. The image size is 1024x768. For similar reasons that we discussed in the previous section, we will use informal subjective evaluation to evaluate our simulation results.

The optimal zooming factor $\hat{\eta}$ is first searched using Eq. (4.15). Based on this optimal zooming factor $\hat{\eta} = 0.97$, the front image I_f is re-scaled to match with I_b and the resulting image is $(\mathcal{H}I_f)(\hat{\eta}\boldsymbol{m})$. Then the disparities between $I_b(\boldsymbol{m})$ and $(\mathcal{H}I_f)(\hat{\eta}\boldsymbol{m})$ are estimated using a region-based approach to obtain an initial estimation using the following steps. Image $I_b(\boldsymbol{m})$ is divided into 16 similar size regions with some overlapped areas between adjacent regions. Fig. 4.10 illustrates the way to generate such overlapped regions (blocks). The image is divided into 16 regular blocks with the same size (for an image with size of 1024x768, the size of each block is 256x192). These blocks are illustrated by the light-dashed lines in the figure. Three types of the overlapped regions (blocks B_1, B_2, B_3 in dark-dashed line) are generated based on the above regular blocks. There are 16 such overlapped regions in total. Generally, the disparities are estimated on these overlapped regions but only the disparities on the correspondent regular block are consider to be valid.

The corresponding region in $(\mathcal{H}I_f)(\hat{\eta}\boldsymbol{m})$ of each region in $I_b(\boldsymbol{m})$ is found through a cross-correlation-based searching. After the disparities between corresponding region pairs have been estimated using optic flow estimation, the disparities between $(\mathcal{H}I_f)(\hat{\eta}\boldsymbol{m})$ and $I_b(\boldsymbol{m})$ can be obtained. We use the method in [100] to calculate dense disparities between the corresponding regions. Some block effects exist in the current disparity maps. Using these disparities as coarse estimations, more accurate disparity maps can be obtained by the proposed algorithm. The disparities between I_f and I_b can then be calculated through the correspondent inverse transformation. Fig. 4.11 shows the disparity maps along both horizontal and vertical directions with this approach. The images of the disparity maps are obtained by mapping the disparity values to image intensities for illustration purpose. We can find that the object structures appear very clear in the disparity maps. For comparison, the disparity maps obtained by the dense disparity estimation method [100] are shown in Fig. 4.12.

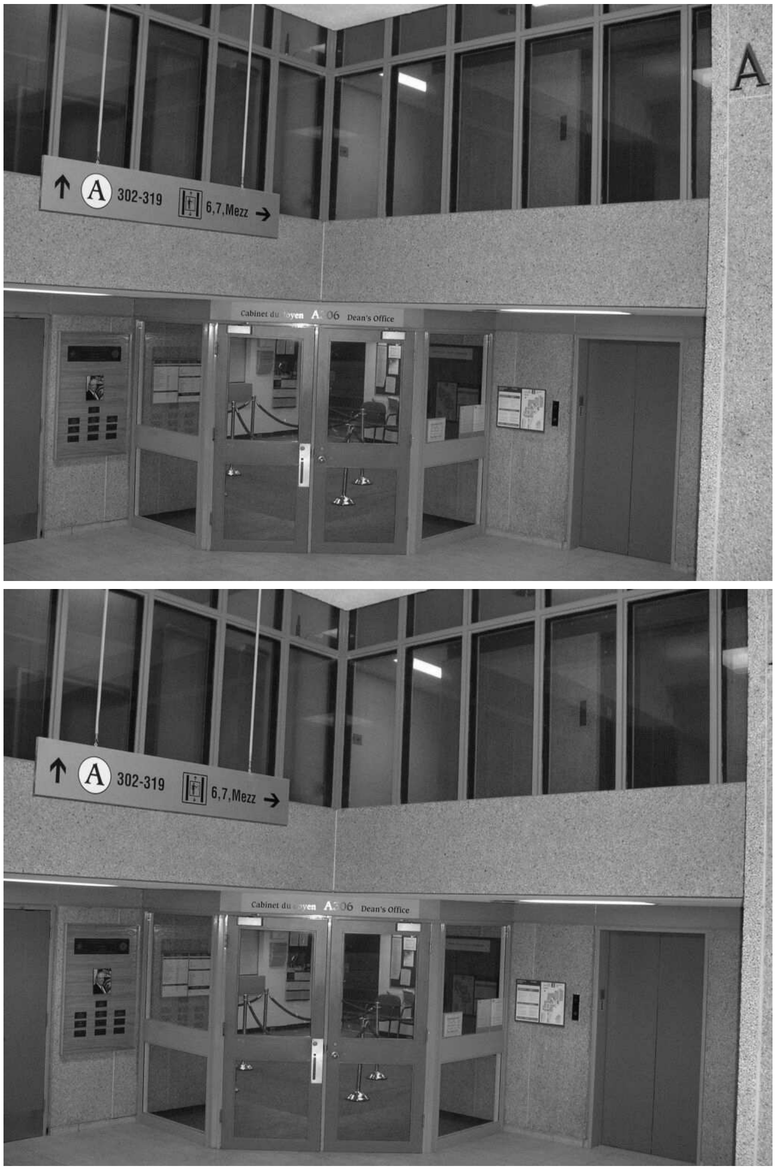


Figure 4.9: The original images

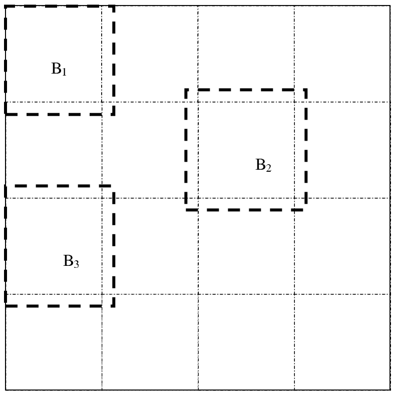


Figure 4.10: three typical regions (dark-dashed blocks B_1, B_2, B_3)

One view with a virtual camera located between the camera positions where I_f and I_b were taken was interpolated is shown at the top part in Fig. 4.13. The interpolated view looks good although there are some defects on the boundary area. The boundary defect is not a significant problem for IBR applications because the overlap areas between pre-captured images can be controlled during acquisition. The approach of using dense matching algorithm directly on the original images [100] has been tested and one of the results is shown at the bottom part in Fig. 4.13 (the circular area shows some artifacts that this algorithm brings). We find that the result generated by the traditional approach is of very poor quality for such situations.

4.3 Chapter summary

In this chapter, two different view interpolation algorithms are proposed for IBR application. Generally, the navigation area can be triangulated by a set of positions, where the pre-captured images are taken, and consequently, a view interpolation algorithm based on three source images has to be used. Thus, a matching-feature-based view interpolation algorithm is proposed to deal with such a general scenario. This leads to our first view interpolation algorithm discussed in this chapter.

In some particular scenarios, the view interpolation has been applied based on

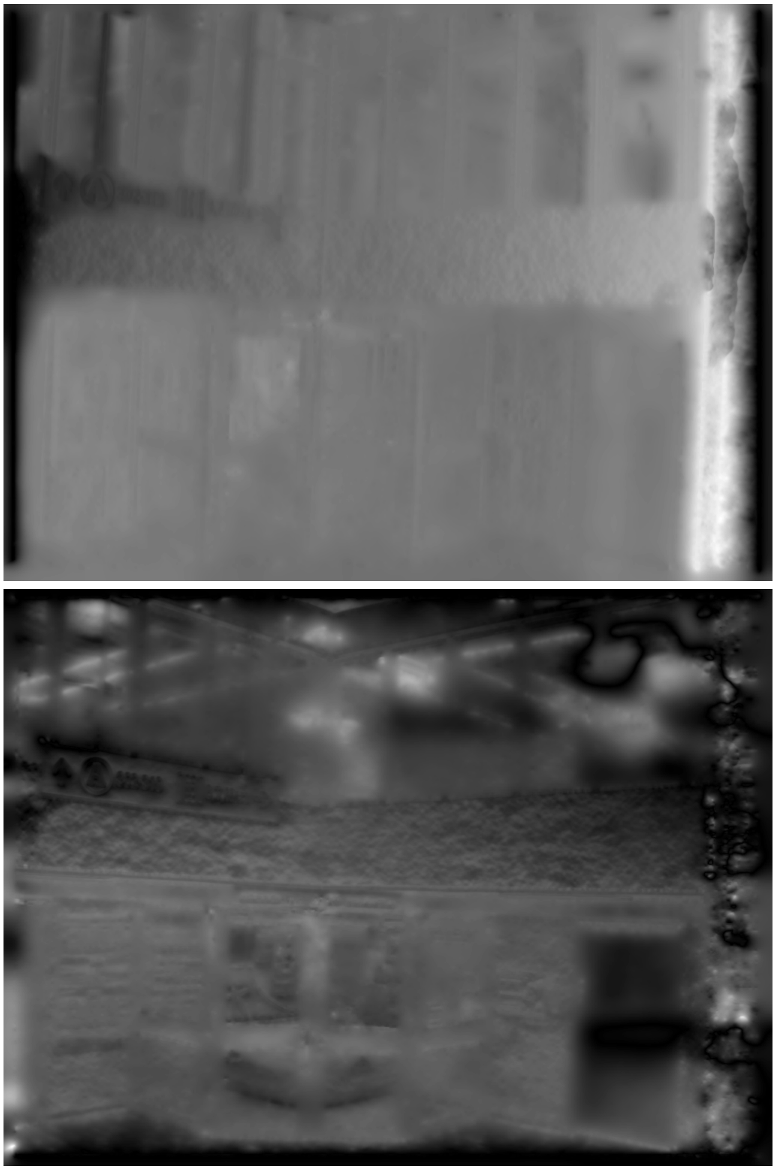


Figure 4.11: Disparity map along horizontal (top) and vertical (bottom) directions obtained from the proposed algorithm

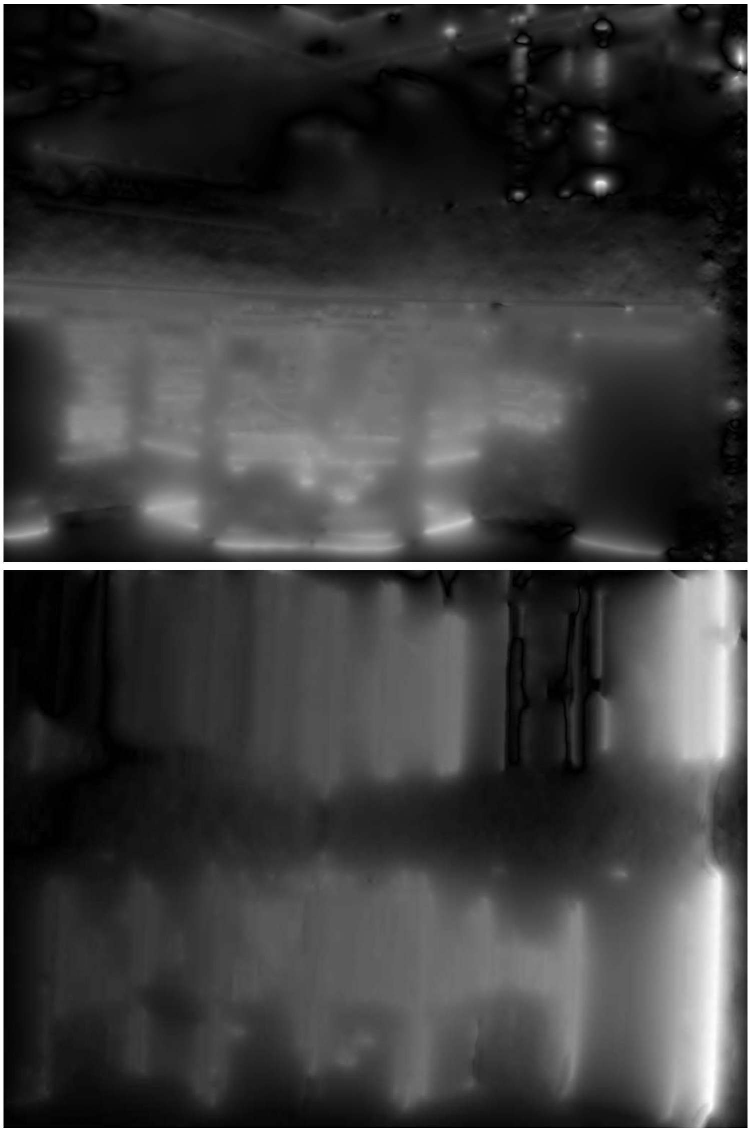


Figure 4.12: Disparity map along horizontal (top) and vertical (bottom) directions obtained from the comparison method

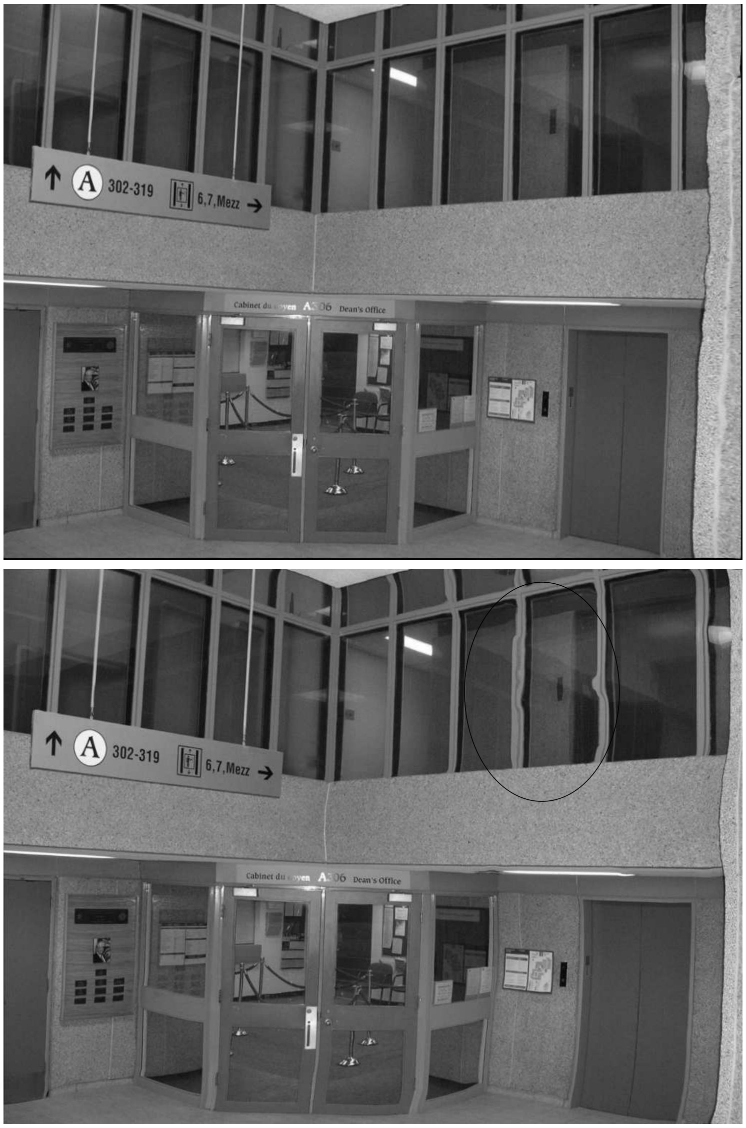

A
302-319
6,7,Mezz
Cabinet du Doyen A306 Dean's Office
A
302-319
6,7,Mezz
Cabinet du Doyen A306 Dean's Office

Figure 4.13: Interpolated in-between view generated by the proposed algorithm (top) and by the comparison method (bottom)

two source images. Previous work that has been published in the literature only considers the situation where two source images are taken side by side. This is not enough for IBR application. The two source images could be taken where the camera moves forward or backward, or even with more general motions. This brings us to our second view interpolation algorithm discussed in this chapter.

In the first view interpolation algorithm, the triangulation of the images and the affine transformation models are used. In this way, feature matchings instead of dense disparities are required because the feature matchings are usually more reliable. A multiple view matching detection strategy is described, which can obtain a large number of approximately uniformly distributed matchings.

In the second algorithm, view interpolation under the special circumstance, the camera moving forward or backward, is described. After studying the physical imaging condition for such a situation, a new disparity estimation model for IBR application is proposed by applying warping on one of the source views. The transformations are specified from physical imaging conditions of camera movement, which can be (but are not limited to) translation, zooming, affine transformation, etc. Simulation results show the proposed method can yield good results compared to applying traditional dense matching algorithm directly on the source views in some scenarios of camera movement such as forward or backward motion.

Chapter 5

Column-based view synthesis: Simplified Concentric Mosaics

In the Light Field description approach, the rendering can be almost scene independent and thus it is reliable, robust and relatively simple. However, a large number of pre-captured images is needed and the technical requirements are relatively expensive due to the precise control of camera movements that is required in the image acquisition phase. The Concentric Mosaics (COM) technique [17] is a clever way to reduce such technical requirements within the class of techniques using the Light-Field description. In order to further reduce the technical requirements for implementation, rendering with non-uniform approximate Concentric Mosaics [108] has been proposed in the scenario that the pre-captured images are obtained by a hand-held camera and the camera positions are estimated through general camera-position estimation techniques of computer vision. However, the precision of such estimates of the camera positions moving in 3D space based on the current techniques in computer vision are usually inadequate to satisfy the requirements of this application, significantly affecting the quality of the rendered novel views.

In this chapter, a simplified Concentric Mosaics technique with non-uniformly-distributed pre-captured images is addressed. Comparing to a similar approach in [109], camera calibration is not required. On the other hand, the constraint that the camera's movement is on a circle, as in the conventional Concentric Mosaics technique, remains, thus providing the potential to considerably simplify the camera-position estimation compared to the method in [108], and also significantly reducing the estimation errors because the camera positions are estimated in a reduced dimension, i.e., one dimension. We call our method the Simplified Concentric Mosaics (SCOM) technique.

5.1 Problem formulation and basic approach

A concentric-mosaics dataset is obtained by recording an image sequence as a camera moves on a circular path, with the camera pointing in the outward radial direction. Typically, the camera is mounted at one end of a long bar while the other end of the bar is pivoted on a tripod or other more stable platform [17]; the bar rotates about this pivot point to capture the image sequence.

The capturing procedure in the Concentric Mosaics technique can be illustrated with Fig. 5.1. The camera is mounted at one end E of the rotation beam CE. When CE rotates around C at a constant velocity, the video camera takes a sequence of images. A set of pre-captured images are obtained after CE has completed one circle. One practical device to implement the Concentric Mosaics technique built by Microsoft Research [110] is shown in Fig. 5.2.

While the rotation can be precisely controlled by a motor to generate a uniform sampling of the circle, we consider here the situation where the bar is (slowly) rotated manually, thus generating a non-uniform sampling. In order to be able to reconstruct the image at arbitrary points of view in the navigation area, the position of the camera at each of the sampling positions must be accurately known.

Assume that N non-uniformly spaced images $I_1, I_2, \ldots, I_N$ pointing in the radial direction are captured on one full revolution over the circular path. The camera position is considered as the camera projection center and is assumed to lie on a circle in a horizontal plane. Typically N is of the order of hundreds, or even several

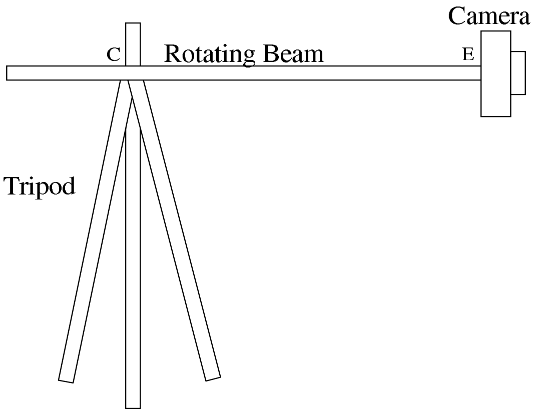


Figure 5.1: The illustration of capture procedure for Concentric Mosaics technique

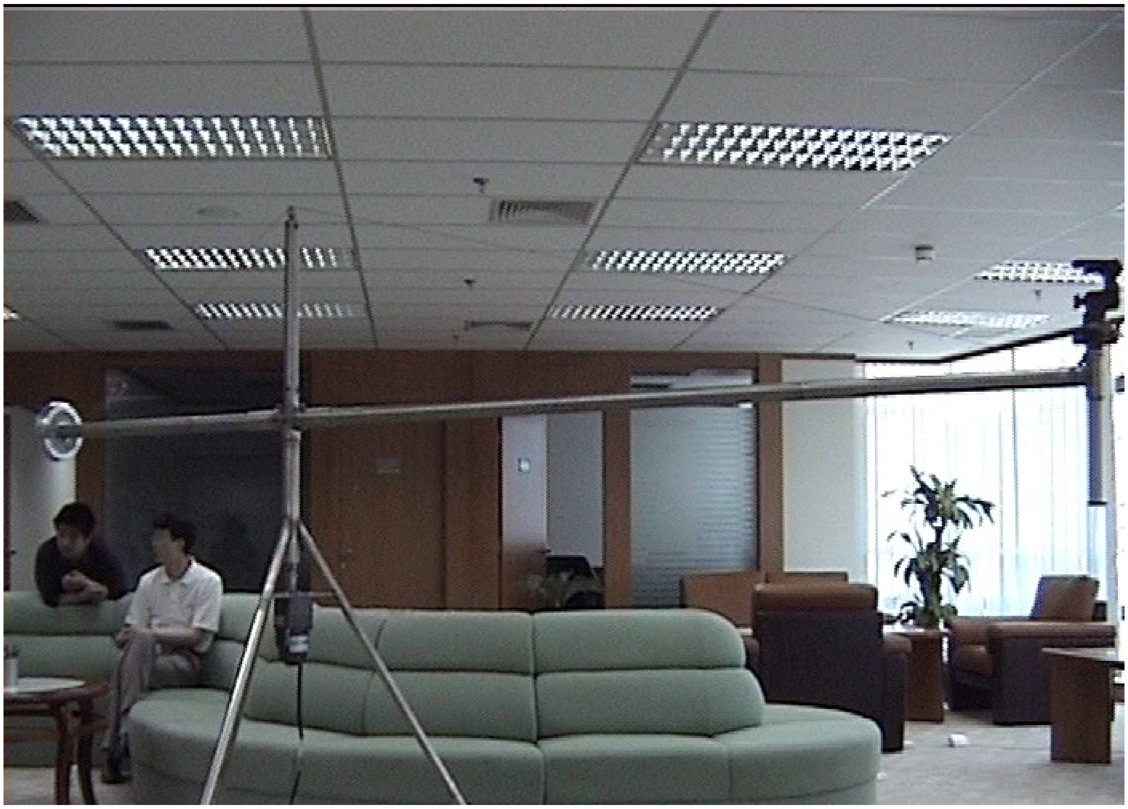


Figure 5.2: The Concentric Mosaics capture device (Microsoft Research)

thousands, of images. The angle between adjacent images I_k and I_{k+1} is denoted θ_k. Then, $\sum_{k=1}^{N} \theta_k = 2\pi$, where we set $I_{N+1} = I_1$. The position of the j^{th} image is given by the cumulative angle with respect to the initial or reference image,

$$\phi_j = \sum_{k=1}^{j} \theta_k \tag{5.1}$$

where $\phi_0 = 0$ and $\phi_N = 2\pi$. The problem is to estimate $\phi_j, j = 1, \ldots, N-1$ from the N captured images.

Our approach is based on matching features in triples of adjacent images, say I_k, I_{k+1} and I_{k+2}. Using standard techniques, we select feature points that can be reliably matched in the three images. From the geometry of projective image formation, we can find an expression for the ratio of the two adjacent angles θ_{k+1}/θ_k in terms of the disparities of these feature points between images I_k and I_{k+1}, and between I_{k+1} and I_{k+2}. Using multiple sets of matching features per image triple, we can then estimate $\eta_k = \theta_{k+1}/\theta_k$ for $k = 1, 2, \ldots$. This leads to a set of linear equations that can be solved for the θ_k and thus the ϕ_k can be estimated.

5.2 Estimation of the camera positions from the non-uniformly-sampled images

Camera position estimation from an image sequence has been extensively studied in computer vision [111]. Assume that an image sequence has been taken by a calibrated camera. By tracking the positions of a number of feature points in the image sequence, the projection matrices can be obtained. The relative camera positions can be calculated as external parameters after decomposing the projection matrices. Then, bundle adjustment is applied in the camera position estimation refinement to get the converged solutions. A large amount of computation is required to obtain the camera position using this technique. Moreover, when the differences between the camera positions are too small, the epipolar constraints are not reliable and the bundle adjustment algorithms that are commonly used in computer vision may not converge in this situation.

Fortunately, the camera position estimation can be greatly simplified due to the special 1D trajectory of the camera movement in SCOM. In this chapter, we propose a simple stereo approach to estimate the camera positions.

5.2.1 Relationship between the angular spacing and disparities on a circular camera path

We use the stereo technique for depth estimation [112], as illustrated in Fig. 5.3. The positions (projection center) of the camera at two adjacent points on a circular trajectory are denoted C_1 and C_2. Images I_1 and I_2 are captured at these two positions. A scene point P_1 is projected at positions $P_{1,1}$ and $P_{1,2}$ in the images I_1 and I_2, respectively. The line segment C_1C_2 is called the baseline for the stereo pair I_1, I_2 and its length is denoted by $b_{1,2}$. The camera coordinate systems (X, Y, Z) at the two camera positions are shown in the figure, with the Y directions pointing out of paper. (X_{1L}, Z_{1L}) and (X_{1R}, Z_{1R}) denote the (X, Z) coordinates of the scene point P_1 in the two camera systems. The angle between the optical axis (in the same plane) of the two cameras is θ. Because the scene point is far away and the angle θ is small, depth values Z_{1L} and Z_{1R} can be approximated by [112]

$$Z_{1L} \approx Z_{1R} \approx \frac{f \cdot b_{1,2}}{d(P_{1,1}, P_{1,2}) - f \cdot \theta} \tag{5.2}$$

where $d(P_{1,1}, P_{1,2}) = X_{1L} - X_{1R}$ is the horizontal disparity between points $P_{1,1}$ and $P_{1,2}$ and f is the focal length of the camera.

We now consider three consecutive views I_1, I_2, and I_3 on a circular arc at positions C_1, C_2, and C_3 respectively, as shown in Fig. 5.4. The angle between C_1 and C_2 is θ_1, and the angle between C_2 and C_3 is θ_2. Assume that the scene point $P_i^{(1)}$ is projected into images I_1, I_2 and I_3 at positions $P_{i,1}^{(1)}$, $P_{i,2}^{(1)}$ and $P_{i,3}^{(1)}$ respectively, $i = 1, 2, \dots M_1$, where M_1 feature points visible in images I_1, I_2, and I_3 have been identified. Applying

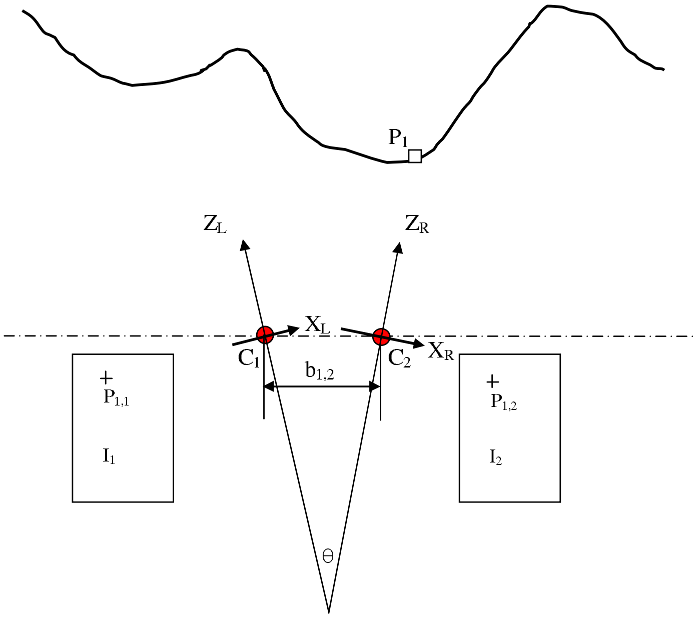


Figure 5.3: Setup for depth estimation using the stereo technique with two views

(5.2) for the image pairs I_1, I_2 and I_2, I_3 respectively, we obtain

$$Z_{i,1L}^{(1)} \approx Z_{i,1R}^{(1)} \approx \frac{f \cdot b_{1,2}}{d(P_{i,1}^{(1)}, P_{i,2}^{(1)}) - f \cdot \theta_1} \tag{5.3}$$

and

$$Z_{i,2L}^{(1)} \approx Z_{i,2R}^{(1)} \approx \frac{f \cdot b_{2,3}}{d(P_{i,2}^{(1)}, P_{i,3}^{(1)}) - f \cdot \theta_2} \tag{5.4}$$

where $Z_{i,1L}^{(1)}$ and $Z_{i,1R}^{(1)}$ are the Z coordinates of scene point $P_i^{(1)}$ in the stereo system with camera positions at C_1 and C_2, and $Z_{i,2L}^{(1)}$ and $Z_{i,2R}^{(1)}$ are the Z coordinates of scene point $P_i^{(1)}$ in the stereo system with camera positions at C_2 and C_3. Image I_2 is the right image for stereo pair (I_1, I_2) and the left image for stereo pair (I_2, I_3), and thus $Z_{i,1R}^{(1)} = Z_{i,2L}^{(1)}$. Because the camera is moving on a circle, we have $\theta_1 \approx b_{1,2}/R$ and $\theta_2 \approx b_{2,3}/R$, where R is the radius of the circle. Substituting these relationships into equations (5.3) and (5.4), we obtain

$$\frac{\theta_1}{d(P_{i,1}^{(1)}, P_{i,2}^{(1)})} \approx \frac{\theta_2}{d(P_{i,2}^{(1)}, P_{i,3}^{(1)})}, \tag{5.5}$$

or equivalently,

$$\theta_2 = \eta_1 \theta_1 \quad \text{where} \quad \eta_1 \approx \frac{d(P_{i,2}^{(1)}, P_{i,3}^{(1)})}{d(P_{i,1}^{(1)}, P_{i,2}^{(1)})}, \quad i = 1, \ldots, M_1. \tag{5.6}$$

For a general consecutive pair of angles, this can be expressed

$$\theta_{k+1} = \eta_k \theta_k \quad \text{where} \quad \eta_k \approx \frac{d(P_{i,k+1}^{(k)}, P_{i,k+2}^{(k)})}{d(P_{i,k}^{(k)}, P_{i,k+1}^{(k)})}, \quad i = 1, \ldots, M_k. \tag{5.7}$$

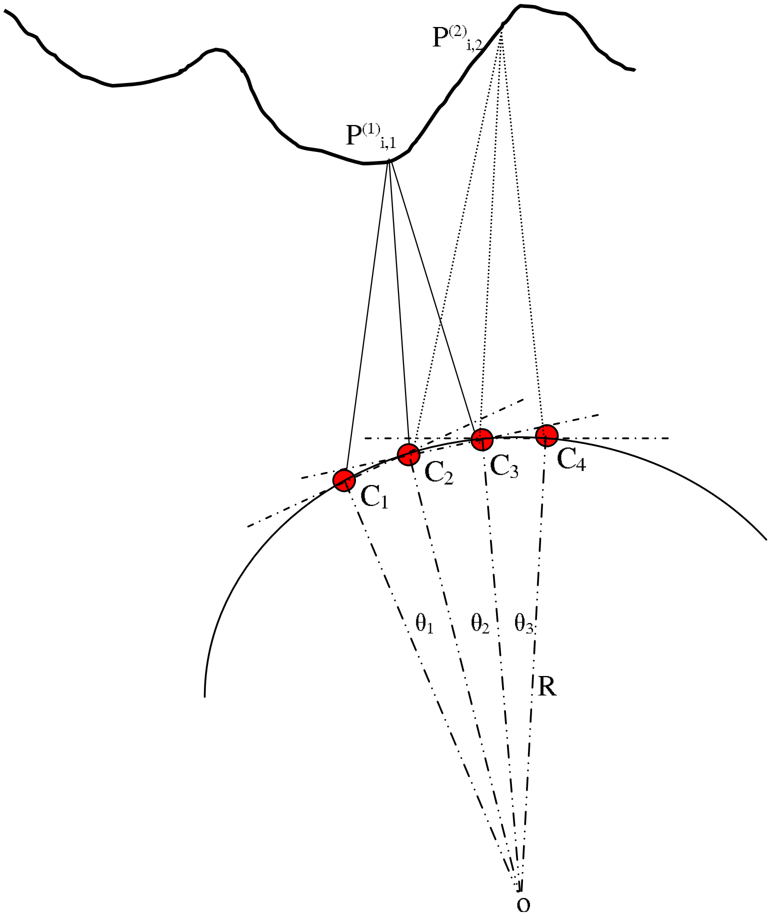


Figure 5.4: Setup for depth estimation using the stereo technique with multiple views

The angle ratio $\eta_k = \theta_{k+1}/\theta_k$ can be estimated using least squares over the M_k

sets of matching features as

$$\hat{\eta}_k = \arg\min_{\eta} \sum_{i=1}^{M_k} (d(P_{i,k+1}^{(k)}, P_{i,k+2}^{(k)}) - \eta d(P_{i,k}^{(k)}, P_{i,k+1}^{(k)}))^2. \tag{5.8}$$

Solving this standard linear least-squares problem [86],

$$\hat{\eta}_k = \frac{\sum_{i=1}^{M_k} d(P_{i,k+1}^{(k)}, P_{i,k+2}^{(k)}) \cdot d(P_{i,k}^{(k)}, P_{i,k+1}^{(k)})}{\sum_{i=1}^{M_k} d(P_{i,k}^{(k)}, P_{i,k+1}^{(k)})^2}. \tag{5.9}$$

Although total least squares may be more appropriate to solve this problem since there are measurement errors in both variables $d(P_{i,k}^{(k)}, P_{i,k+1}^{(k)})$ and $d(P_{i,k+1}^{(k)}, P_{i,k+2}^{(k)})$, this approach was tested and did not provide any significant improvement. So it was not used further.

Matching errors are unavoidable and play a very significant role in the proposed stereo-based camera position estimation method. Reliable methods such as RANSAC [113] have been developed to robustly solve such problems by discounting outliers due to faulty matches. Under this special circumstance, we use a method we call ratio-fitting to select good matching features from the matching feature pool. Because we know that the disparity ratio between two adjacent stereo pairs should be a constant, we select the majority of matching features that respect this constant-ratio principle as good matching features. The method can be illustrated using the above set, denoted as Π_k, of M_k sets of matching features. For each pair of disparities $d(P_{i,k+1}^{(k)}, P_{i,k+2}^{(k)})$ and $d(P_{i,k}^{(k)}, P_{i,k+1}^{(k)})$, the individual disparity ratio is calculated as $\eta_{i,k} = d(P_{i,k+1}^{(k)}, P_{i,k+2}^{(k)})/d(P_{i,k}^{(k)}, P_{i,k+1}^{(k)})$. According to the proposed theory, all $\eta_{i,k}$ should be a constant in the ideal situation, which is usually not the case due to matching errors. The mean of $\eta_{i,k}$, denoted as $\overline{\eta}_k$, is calculated. We define a confidence factor χ. Set Π_k is separated into two subsets, subset Π_k^g containing good matching features and subset Π_k^f containing faulty matching features, with $\Pi_k^g \cap \Pi_k^f = \emptyset$ and $\Pi_k^g \cup \Pi_k^f = \Pi_k$, where $\emptyset$ represents the empty set. There are χM_k matching features in Π_k^g and $(1-\chi)M_k$ matching features in Π_k^f. For all $\eta_{p,k} \in \Pi_k^g$ and for all $\eta_{q,k} \in \Pi_k^f$, $|\eta_{p,k} - \overline{\eta}_k| < |\eta_{q,k} - \overline{\eta}_k|$. (The confidence factor $\chi = 0.95$ was used in our experiments.)

Then Π_k is updated by Π_k^g while M_k is updated to χM_k ($M_k = \chi M_k$). It is straightforward that the ratio-fitting algorithm can be applied on updated Π_k and M_k with the same (or a different) confidence factor χ. Thus, the above ratio-fitting method to find good matching features can be implemented in an iterative way. The idea here is similar to the one used in vector quantization [114].

Moreover, because the camera motions are almost in the horizontal plane and the range of the disparities between matching features can be roughly estimated due to the regular motion of the camera, the search window can be consequently specified. In Fig. 5.5, points $P_{1,1}$ and $P_{1,2}$ are a pair of matching features in image I_1 and image I_2. The searching window (the dashed area in image I_2) can be relatively small, and narrow in the vertical direction, for the above camera motion scenario. This greatly reduces the number of possible faulty matching features.

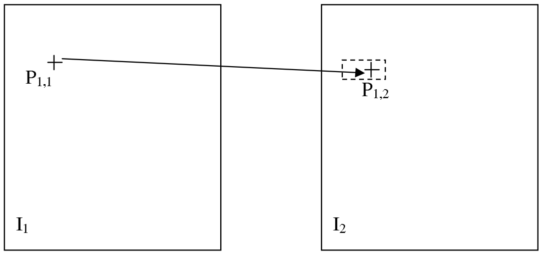


Figure 5.5: Illustration of searching window to detect matching features in the SCOM.

Although some approximations have been made to arrive at the fundamental relation (5.7) that relates the ratio of adjacent angles to the ratio of horizontal disparities of corresponding points, we expect these to be minor in normal circumstances and for errors to be mainly due to imprecision in measuring the disparities. To verify the linear relationship (5.7), we carried out a test with the concentric mosaics dataset ‘kids’ provided by Microsoft Research [17]. In this dataset, the images were acquired using a precision motor and thus we assume that the images are uniformly distributed over the circle. The sequence consists of $N = 2967$ images that we assume are equally spaced with angular separation $2\pi/2967$ rad. The sequence was subsampled by 15 to get a uniform sequence with angular spacing of $2\pi/198$ rad or about $1.82°$. Triples of matching features were detected using the projective vision toolkit (PVT) [84] (the proposed ratio-fitting technique is not applied to remove outliers). Fig. 5.6(a) shows a scatterplot of the corresponding pairs of disparities $(d(P_{i,k}^{(k)}, P_{i,k+1}^{(k)}), d(P_{i,k+1}^{(k)}, P_{i,k+2}^{(k)}))$ over a set of 194 image triples, where there is a variable number M_k of feature sets per image triple (mean = 39.8 , standard deviation = 13.8). It is clear visually that the trend is linear; the least-squares line with $\eta = 0.997$ is also shown. Fig. 5.6(b) shows a similar scatter plot for one particular triple of images. Again the trend is linear and the least-squares line with $\hat{\eta} = 0.958$ is shown. Over the entire set of image triples, we obtain mean$(\hat{\eta}_k) = 0.998$ and std$(\hat{\eta}_k) = 0.035$.

This experiment was repeated with different angular spacings (namely 1, 4, 8), and the results are summarized in Table 5.1. In this table, the means and standard deviations of the estimated angle ratios are given for the different angular spacings. The means and standard deviations of the number of matching features that were used for the correspondent estimations are also given for different angular spacings. The linear trend was observed in all cases, but the bias is seen to decrease as the angle increases, since the relative importance of the disparity errors decreases. In particular, the angular spacing of $2\pi/2967$ has the largest bias and standard deviation of the error. The explanations will be given in the following section (Section (5.3.1): analysis of the ratio estimation errors). As a consequence, the angle grouping technique will be proposed in the subsequent section (Section (5.3.2)).

Table 5.1: experimental data on angle ratio: for different angular spacings, the means and standard deviations of the estimated angle ratios, the means and standard deviations of the number of matching features used for the correspondent estimations

angular spacing	estimated angle ratio		number of matches	
	mean	std	mean	std
1	0.873	0.103	18.5	4.1
4	0.981	0.026	52.7	15.0
8	0.992	0.028	53.7	15.8
15	0.998	0.035	39.8	13.8

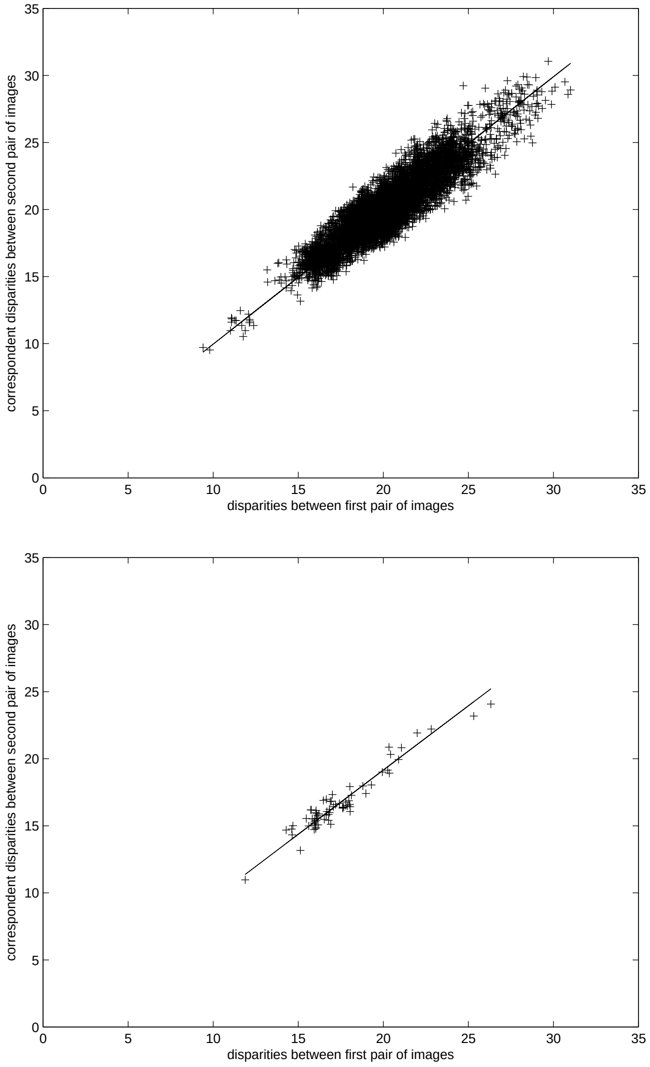


Figure 5.6: Scatterplots of pairs of disparities. (a) Over a set of 194 image triples. (b) For one specific image triple.

5.2.2 Estimation of the camera positions on a circle

From equation (5.7), and with the estimated values of $\hat{\eta}_k$ from equation (5.9), we obtain

$$\theta_2 = \hat{\eta}_1 \theta_1 \tag{5.10}$$

$$\theta_3 = \hat{\eta}_2 \theta_2 = \hat{\eta}_1 \hat{\eta}_2 \theta_1 \tag{5.11}$$

and in general

$$\theta_{k+1} = \left(\prod_{i=1}^{k} \hat{\eta}_i \right) \theta_1, \quad k = 1, 2, \ldots, N-1. \tag{5.12}$$

The one remaining unknown θ_1 can be obtained by imposing the constraint $\sum_{k=1}^{N} \theta_k = 2\pi$ to yield

$$\theta_1 = \frac{2\pi}{\sum_{k=1}^{N} \left(\prod_{i=1}^{k} \hat{\eta}_i \right)}, \tag{5.13}$$

and from there we can compute the ϕ_j using equation (5.1).

Although these N linear equations in N unknowns will give an exact solution if the correct values of η_i are used, the estimation errors in the $\hat{\eta}_i$ will cause increasing errors in the θ_k, as can easily be appreciated from equation (5.12), and these errors will be further accumulated in the computation of the ϕ_j. Since the circle constraint assures that ϕ_N will be 2π, the largest errors are found in the center of the range. However, we note that one additional independent constraint can be applied, namely

$$\theta_1 = \eta_N \theta_N, \tag{5.14}$$

applying a constraint to the ratio of the last angle to the first. By adding this constraint, we obtain an overdetermined set of $N+1$ equations in N unknowns

$$\mathbf{A} \boldsymbol{\theta} = \mathbf{g} \tag{5.15}$$

that can be solved in the least-squares sense. The solution for the individual angles and cumulative angles can then be expressed in matrix notation as

$$\boldsymbol{\theta} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{g}, \tag{5.16}$$

$$\boldsymbol{\phi} = \mathbf{C} \boldsymbol{\theta}, \tag{5.17}$$

where $\boldsymbol{\phi} = \begin{bmatrix}\phi_1 & \phi_2 & \dots & \phi_{N-1}\end{bmatrix}^T$, $\boldsymbol{\theta} = \begin{bmatrix}\theta_1 & \theta_2 & \dots & \theta_N\end{bmatrix}^T$, $\boldsymbol{g} = \begin{bmatrix}0 \ 0 \ ... \ 0 \ 2\pi\end{bmatrix}^T$,

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 & 0 \\ & & & \vdots & & \\ 1 & 1 & 1 & \dots & 1 & 0 \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} \hat{\eta}_1 & -1 & 0 & 0 & \dots & 0 & 0 \\ 0 & \hat{\eta}_2 & -1 & 0 & \dots & 0 & 0 \\ & & & & \vdots & & \\ 0 & 0 & 0 & 0 & \dots & \hat{\eta}_{N-1} & -1 \\ -1 & 0 & 0 & 0 & \dots & 0 & \hat{\eta}_N \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 \end{bmatrix} \tag{5.18}$$

5.3 Obtaining a stable numerical solution

The method described in Section 5.2.2 involves the solution of equation (5.16) for N on the order of 3000. It is not suitable if the errors of the coefficients in matrix A, which are mainly due to the relative errors in the disparity estimates of the feature points, are becoming large.

5.3.1 Analysis of the ratio estimation errors

Considering the estimation errors in the disparity estimate, a more precise representation for the equation $\eta_1 \approx \frac{d(P^{(1)}_{i_1,2}, P^{(1)}_{i_1,3})}{d(P^{(1)}_{i_1,1}, P^{(1)}_{i_1,2})}$ should be

$$\eta_1 \approx \frac{d(P^{(1)}_{i_1,2}, P^{(1)}_{i_1,3}) + \Delta d(P^{(1)}_{i_1,2}, P^{(1)}_{i_1,3})}{d(P^{(1)}_{i_1,1}, P^{(1)}_{i_1,2}) + \Delta d(P^{(1)}_{i_1,1}, P^{(1)}_{i_1,2})}, \tag{5.19}$$

where the $\Delta d(P^{(1)}_{i_1,2}, P^{(1)}_{i_1,3})$ and $\Delta d(P^{(1)}_{i_1,1}, P^{(1)}_{i_1,2})$ are estimation errors on $d(P^{(1)}_{i_1,2}, P^{(1)}_{i_1,3})$ and $d(P^{(1)}_{i_1,1}, P^{(1)}_{i_1,2})$, respectively. The disparity errors depend on the precision of the positions of the detected matching features but are almost independent of the disparity values. In this situation, the variance of the ratio estimate η varies inversely as the variance of the disparity values. For very small angles, and correspondingly small disparities, this disparity variance will be smaller than for larger angles having larger disparities. We found that better angle-ratio estimates could be obtained with disparities of the order of about 20 pixels, or angles in the vicinity of 2°; this gives a sufficiently large disparity variance without compromising the small angle approximations required for equation (5.7). Thus if we can choose the matching features in a way that $d(P^{(1)}_{i_1,1}, P^{(1)}_{i_1,2}) \gg \Delta d(P^{(1)}_{i_1,1}, P^{(1)}_{i_1,2})$ and $d(P^{(1)}_{i_1,2}, P^{(1)}_{i_1,3}) \gg \Delta d(P^{(1)}_{i_1,2}, P^{(1)}_{i_1,3})$, the ratios should be more reliable and accurate.

In the proposed SCOM technique, the camera positions between two adjacent images are very close, so the disparities between the matching features of two adjacent images are usually very small.

5.3.2 Angle grouping technique

Based on the above reasoning, we perform the ratio estimation on grouped angles in order to increase rotation-angle estimation precision. A set of grouped angles can be obtained by grouping L successive angles, or

$$\Theta_{i,j} = \sum_{k=0}^{L-1} \theta_{i+(j-1)*L+k}, \quad i = 1, 2, ..., L; j = 1, 2, ..., N^{'} \tag{5.20}$$

for the j-th grouped angle in the i-th set. There are L such sets where each set has $N^{'} = \lceil \frac{N}{L} \rceil$ grouped angles ($\lceil x \rceil$ means rounding x to the nearest integer not less than x). The index for the rotation angles are interpreted modulo N, or $N + i \equiv i$. As a consequence, we can calculate N' grouped angles in each set using equation (5.16) as described in Section 5.2.2.

After arranging the rotation angles and grouped angles into vector format, we can easily establish the relationship

$$\boldsymbol{B\theta} = \boldsymbol{\Theta} \tag{5.21}$$

where $\boldsymbol{\Theta} = \begin{bmatrix}\Theta_{1,1} & \Theta_{1,2} & ... & \Theta_{1,L} & \Theta_{2,1} & \Theta_{2,2} & ... & \Theta_{2,L} & ... & \Theta_{N',L}\end{bmatrix}^T$ and $\mathbf{B}$ is a matrix consisting of ones and zeros as determined by equation (5.20).

Equation (5.21) contains N unknowns and $N^{'} \cdot L$ equations ($N^{'} \cdot L \leq N$). The solution for $\boldsymbol{\theta}$ is not unique if $N^{'} \cdot L < N$. In addition, we found that the solution for $\boldsymbol{\theta}$ is very unstable if we solve equation (5.21) directly or using the least square method even under the condition of $N^{'} \cdot L = N$. This is because N is usually very large (the order of 3000) and thus the estimation errors in grouped angles $\mathbf{\Theta}$ have significant affect on the solutions for $\boldsymbol{\theta}$. Thus, the solutions for $\boldsymbol{\theta}$ using constrained least squares method and total least squares method will be used in the following section. As a consequence, the condition $N^{'} \cdot L < N$ is no longer an issue, which prevents us from obtaining a unique solution for $\boldsymbol{\theta}$.

Another option is to obtain the position information for only one of the sets of grouped angles. Since the camera acceleration is small, we can assume that the velocity is nearly constant over each grouped angle of about 2° and simply interpolate the intervening positions linearly.

5.3.3 Solution of the linear equations with noisy disparity estimates

Due to the large N (number of angles), the rotation angles in equation (5.21) are usually very small and thus the solutions are very sensitive to the noisy matrix $\mathbf{\Theta}$, even though the angle grouping technique can somewhat relieve the problem caused by the noisy estimation data of the disparities. Some other methods, such as the steepest descent algorithm and conjugate gradient algorithm, can be used. The following two methods are suggested and have been tested to obtaining stable solutions of equation (5.21).

The first method is using the constrained least squares (CLS) method by adding a regularization constraint, which is typically used in image restoration [115]. The solutions are obtained by

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \{ \|\boldsymbol{B\theta} - \mathbf{\Theta}\|^2 + \|\alpha \cdot T\boldsymbol{\theta}\|^2 \} \tag{5.22}$$

where T is a matrix which applies a constraint on local rotation angle. It can be formed by some high-pass filters, such as the filter coefficients [-0.5 1 -0.5]. α is a regularization factor. The minimization problem can be solved by

$$(\boldsymbol{B}^T\boldsymbol{B} + \alpha T^T T)\theta = \boldsymbol{B}^T\mathbf{\Theta}. \tag{5.23}$$

The second method is using the total least squares (TLS) method to obtain stable solutions of the large scale set of linear equations. In equation (5.15), matrix $\boldsymbol{A}$ is noisy; in equation (5.21) matrix $\boldsymbol{\Theta}$ is noisy. We use $\boldsymbol{\Phi}\boldsymbol{\theta} = \boldsymbol{\Gamma}$ to represent these two equations, where in general, both $\boldsymbol{\Phi}$ and $\boldsymbol{\Gamma}$ may contain errors. We are looking for a stable solution of the equation

$$(\boldsymbol{\Phi} + \Delta\boldsymbol{\Phi})\boldsymbol{\theta} = \boldsymbol{\Gamma} + \Delta\boldsymbol{\Gamma} \tag{5.24}$$

to estimate the solution of the original equations $\boldsymbol{\Phi}\boldsymbol{\theta} = \boldsymbol{\Gamma}$ with minimal $\|\Delta\boldsymbol{\Phi}\|_F$ and $\|\Delta\boldsymbol{\Gamma}\|_F$. $\|\cdot\|_F$ denotes Frobenius norm of a matrix. $\Delta\boldsymbol{\Phi}$ and $\Delta\boldsymbol{\Gamma}$ are the error matrices. The Frobenius norm of matrix $\Delta\boldsymbol{\Phi}$ is defined as

$$\|\Delta\boldsymbol{\Phi}\|_F = (\sum_i \sum_j |\Delta\boldsymbol{\Phi}(i,j)|^2)^{1/2} \tag{5.25}$$

and similarly for $\|\Delta\boldsymbol{\Gamma}\|_F$. An algorithm to solve the TLS problem using singular value decomposition is well known and further details can be found in [116].

There is no analytical method to determine the regularization factor because of a lack of knowledge about the noise, or disparity estimation errors. A suitable regularization factor for this application can be obtained through observing the distribution of rotation angles. During the acquisition of pre-captured images using SCOM, we try to rotate the long beam as slowly as possible, and also try to keep the rotation velocity as a constant. As a consequence, we know that the rotation angle should be smoothly changing. Thus, the regularization factor can be selected that gives a reasonably smooth solution.

5.3.4 Summary on solution of the linear equations

In this chapter, one major task is to find the stable solution of the linear equations. For different situations, we have introduced different methods for solution such as least-squares method, constrained least-squares method and total least-squares method. If the number of unknown angles is on the order of 300 (or least), the least-squares method can give good enough solution. The small number of unknown angles means the large angular spacing. This is the scenario of obtaining the grouped angles ($\mathbf{\Theta}$ in equation (5.21)). When we solve equation (5.21), the number of unknown angles is on the order of 3000 (or more). Then, the constrained least-squares method or the total least-squares method has to be used to obtain stable solutions.

5.4 Pre-processing of the pre-captured images

In the above discussion, we assume that there are no vertical disparities between any two of the pre-captured images. However, it may not be the case in practice. The view rendering in the Concentric Mosaics technique is a column-based view interpolation. Thus, the vertical motions, and any other motions that deviate from the ideal ones, in the pre-capturing procedure will significantly affect the quality of the synthesized images.

In this section, we give a method to eliminate or at least reduce the possible vertical offsets and distortions in the pre-captured images. Because the distance between the camera positions where two adjacent images were taken is very short and the camera is pointing outward along the long bar, the two adjacent images are captured approximately in parallel directions. The view rectification technique of computer vision can be used.

In this chapter, affine transformations are used to reduce such offsets and distortions. The correspondent camera motions may generally include rotations around its center in the vertical plane and vertical movements. Thus the approach is similar to what we proposed in [79].

Affine transformations are originally defined on continuous-space images. For a discrete image I, the affine transformation is defined as follows. Assume that a continuous image corresponding to I can be obtained using a linear interpolation operator $\mathcal{H}$. Then the discrete affine transformation operator with parameter $\boldsymbol{t}$, denoted $\mathcal{A}_{\boldsymbol{t}}$, is defined by $(\mathcal{A}_{\boldsymbol{t}}I)(\boldsymbol{x}) = (\mathcal{H}I)(\boldsymbol{T}(\boldsymbol{x}-\boldsymbol{x}_0)+\boldsymbol{d}+\boldsymbol{x}_0), \boldsymbol{x} \in \Lambda$. Thus, there are six parameters to define an affine transformation, namely the vector $\boldsymbol{t} = [t_{11} \;\; t_{12} \;\; t_{21} \;\; t_{22} \;\; d_1 \;\; d_2]^T$, where d_1 and d_2 denote the translations along horizontal and vertical direction respectively, $\boldsymbol{T} = \left[\begin{smallmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{smallmatrix}\right]$ and $\boldsymbol{d} = [d_1 \quad d_2]^T$. $\boldsymbol{x}_0$ is a reference point based on which the transformations are applied, such as the image center. Λ is the sampling lattice where image $(\mathcal{A}_{\boldsymbol{t}}I)(\boldsymbol{x})$ is defined. Of course, $(\mathcal{H}I)(\boldsymbol{x}^{'})$ is only computed at the points $\boldsymbol{x}^{'} = \boldsymbol{T}(\boldsymbol{x} - \boldsymbol{x}_0) + \boldsymbol{d} + \boldsymbol{x}_0, \boldsymbol{x} \in \Lambda$, using any suitable interpolation such as bilinear, bicubic, spline, etc.

Assume that $I_1, I_2, ..., I_N$ are the pre-captured images. For any image I_i, the optimal global affine transformation is first found as

$$\hat{\boldsymbol{t}}_i = \arg\min_{\boldsymbol{t}_i} \sum_{\boldsymbol{x}} |(\mathcal{A}_{\boldsymbol{t}_i} I_i)(\boldsymbol{x}) - \widetilde{I}_{i-1}(\boldsymbol{x})| \tag{5.26}$$

where, $\widetilde{I}_{i-1} = (\mathcal{A}_{\hat{\boldsymbol{t}}'_{i-1}} I_{i-1})(\boldsymbol{x})$ and $\widetilde{I}_1 = I_1$. The parameter vector $\hat{\boldsymbol{t}}'_{i-1}$ is given by

$$\hat{\boldsymbol{t}}'_{i-1} = \Big[t_{11}(i-1) \;\; t_{12}(i-1) \;\; t_{21}(i-1) \;\; t_{22}(i-1) \;\; 0 \;\; d_2(i-1) \Big]^T \tag{5.27}$$

if

$$\hat{\boldsymbol{t}}_{i-1} = \Big[t_{11}(i-1) \;\; t_{12}(i-1) \;\; t_{21}(i-1) \;\; t_{22}(i-1) \;\; d_1(i-1) \;\; d_2(i-1) \Big]^T \tag{5.28}$$

The above pre-processing is applied for $i = 2, 3, ..., N$. If the vertical disparities are the only issue of concern, the rotation matrix will be an identity matrix. The method has been tested on a sequence taken by a video camera and the camera movement was slightly fluctuated in the vertical direction on the SCOM setup.

5.5 Rendering with irregular image samples

The capturing and rendering procedure of the Concentric Mosaics technique is abstractly illustrated in Fig. 5.7. The camera is mounted on one end of the long rotation bar CE. When CE rotates around C at a constant velocity, the video camera takes images. The navigation area is the area within the inner dashed circle shown in Fig. 5.7, within which any arbitrary views can be synthesized through the Concentric

Mosaics rendering algorithm. The radius r_{NA} of the dashed circle is,

$$r_{\mathrm{NA}} = R \cdot \sin(\frac{\delta_c}{2}) \tag{5.29}$$

where R is the effective length of the rotation beam, which is the distance from the rotation center to the position of the camera on the beam, or CE in the figure and the angle δ_c is the camera's horizontal field of view.

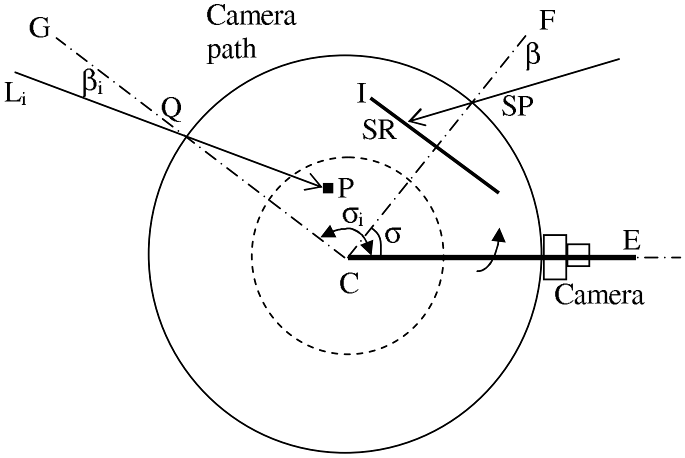


Figure 5.7: The capturing and rendering procedure

In the Concentric Mosaics technique, the pixels in one column of the pre-captured images are grouped into one *condensed-light-ray*, namely a *sampled-ray*. (The word "light-ray" has lost its physical meaning here) The positions at which the images are captured on the camera path are sampled points. In Fig. 5.7, SP is a sampled point and SR is a sampled-ray, which corresponds to one column in the image I taken by the camera at the rotation angle σ. The sampled-ray SR corresponds to a condensed-light-ray through angle β with CF, where CF is perpendicular to image I and passes through its center.

By stacking the pixels of each column into one element, the pre-captured image has a one-dimensional data structure. Thus the data structure of the whole set of pre-captured images can be represented in a σ-β plane as shown in Fig. 5.8. Each dot in Fig. 5.8 denotes a column within the entire set of pre-captured images. All dots in the same horizontal row correspond to one pre-captured image.

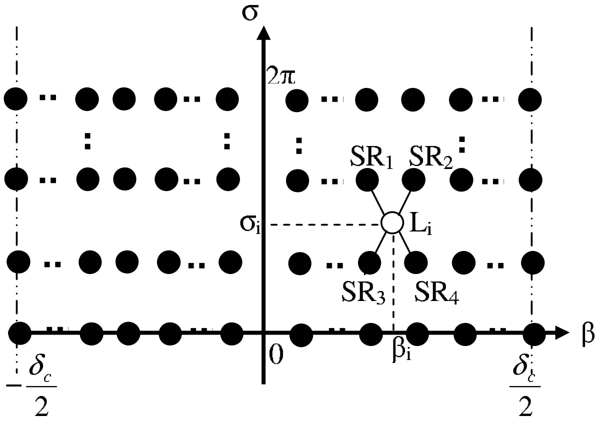


Figure 5.8: The data structure of pre-captured images in σ-β plane for the Concentric Mosaics technique

The rendering procedure can also be illustrated with Fig. 5.7. P is an arbitrary position within the navigation area (the dashed circle) and L_i is one condensed light ray toward P. An arbitrary view at position P is constructed by a set of condensed-light-rays when the viewing direction is given, just like putting a virtual camera at P.

The condensed-light-ray L_i is determined by two angles σ_i and β_i as shown in Fig. 5.7. If the intersection point Q of L_i with the camera path happens to be a sampled point and there is a sampled-ray corresponding to β_i, that sampled-ray can be directly put into the final image. However, that is not generally true. For a general case, L_i is one point illustrated in Fig. 5.8. L_i will be interpolated from its nearby sampled-rays SR_1, SR_2, SR_3 and SR_4 in the figure as

$$L_i = \omega_1 SR_1 + \omega_2 SR_2 + \omega_3 SR_3 + \omega_4 SR_4, \tag{5.30}$$

where ω_1, ω_2, ω_3, and ω_4 are weights for interpolation.

Depth information of the environment, which is difficult to obtain, is required to correctly compute the weights for interpolation [117]. Thus the infinite depth assumption and the constant depth assumption are used; further details can be found in [118]. The nearest point approximation can also be classified in the above interpolation formula, with only one weight equal to one while the others are zero.

In the proposed SCOM technique, the data structure of image samples is very similar to the standard Concentric Mosaics technique and thus so are the rendering methods. The only difference is that the dots, which represent sampled-rays, are no longer uniformly distributed along the σ direction, but are irregularly distributed as shown in Fig. 5.9. However, this will cause little difference in rendering. Deeper discussions on rendering with non-uniform approximate concentric mosaics can be found in [108].

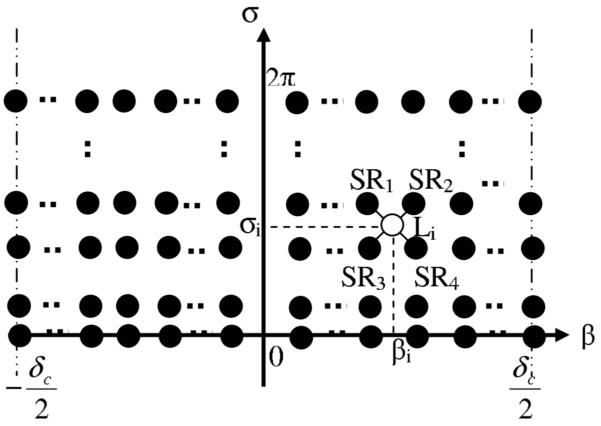


Figure 5.9: The data structure of pre-captured images in σ-β plane for the SCOM technique

5.6 Experimental results

The first two simulations to test the proposed camera-position estimation algorithm were carried out using Concentric Mosaics data (the ‘kids’ sequence in [17]) that was provided by Microsoft Research. In the Concentric Mosaics technique, the camera rotation is controlled by a motor and the camera positions are supposed to be uniformly distributed along the circle. We have no knowledge of the precision of the motor controller that was used by Microsoft Research. We assume that the camera positions are uniformly distributed on the circle and use the set of data as a benchmark to verify the proposed algorithm. Thus, we have ground truth to evaluate our estimation results in these two simulations.

In the first simulation, camera positions which are non-uniformly distributed on a circle were estimated and compared with the ground truth. The purpose is to test the relationship between the disparity ratios of matching features and the rotation angles. This relationship is the fundamental principle used in the proposed camera rotation-angle estimation algorithm. In the second simulation, the camera positions were estimated using the Concentric Mosaics data (the 'kids' sequence). All estimation results of 15 groups are provided for evaluation of the estimation precision of the proposed algorithm. It should be noted that the sets of linear equations involved are directly solved without any other regularization except the closed-loop constraint in the above experiments. Thus, it gives reliable tests on the proposed principle.

The dense rotation angles were then estimated using both the constrained least squares method and total least squares method using the above estimated 15 sets of group angles. The third simulation involved rendering novel views using the proposed SCOM technique for evaluation. In all the simulations, the initial matching features were found using the publicly available PVT software [84], [119].

5.6.1 Simulation I: Test of the proposed camera-position estimation algorithm

In order to test the proposed camera position estimation algorithm in the non-uniform camera position scenario, we selected a sub-sequence of images from the 'kids' sequence in the following way. There are 2967 images in the original sequence. The first 50 images of the sub-sequence were spaced by 8 images, the second 50 images were spaced by 16 images, the third 50 images were spaced by 24 images, and the

Table 5.2: The RMSE value and relative error of estimated angles in each individual segment

Segment number	spacing of images	ground truth	RMSE	relative error
1	8	0.017(rad)	0.00073(rad)	4.28%
2	16	0.034(rad)	0.00115(rad)	3.38%
3	24	0.051(rad)	0.00158(rad)	3.10%
4	8	0.017(rad)	0.00059(rad)	3.49%

remaining images were spaced by 8 images again, resulting in four segments. Assuming the positions in the original sequence were uniform, the angles between adjacent images in these segments was 0.017 rads, 0.034 rads, 0.051 rads and 0.017 rads, respectively. Fig. 5.10 shows the simulation results using the proposed method with the closed-loop constraint and the ratio fitting criterion (only apply ratio fitting criterion once with $\chi = 0.95$). The solutions are obtained by least square method (Equation (5.16)) because the angular spacings in the simulation are large enough. The cumulative angles are shown in the top of the figure. The cumulative angle is defined as the angle between the camera position where the current image was taken and the camera position where the first image was taken. The individual angles between two adjacent camera positions are shown in the bottom of the figure. The solid line is the ideal case, assuming the original pre-captured images are uniformly distributed, and the dashed line shows angles obtained using the proposed camera angle estimation algorithm. The RMSE (root mean square error) of the estimated angles is 0.0010440 rads. The RMSE and relative error of estimated angles in each segment is given in Table 5.2. This simulation shows that the relationship between the disparity ratios of matching features and the rotation angles derived from the theoretical analysis is correct. In addition, we find that this relationship is still valid even though the angles between adjacent images are relatively large.

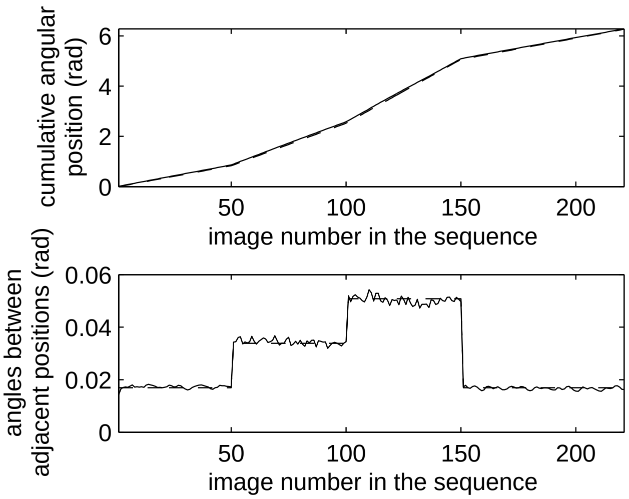


Figure 5.10: The results for simulation I using the proposed method: cumulative angular position (top) and angles between adjacent camera positions (bottom)

5.6.2 Simulation II: camera-position estimation using practical concentric mosaics data

In the second simulation, we tested the proposed camera position estimation algorithm using the full concentric mosaics data as a benchmark. Using the technique in Section (5.3.2), the grouped angles with $L = 15$ (number of groups) were estimated first. For a total number of 2967 images in the sequence, there are 198 angles in each group. Fig. 5.11 shows the distributions of estimated angles in each group. The RMSE values of the estimated angles and the relative errors in each group are given in Table 5.3. Here, equation (5.15) was solved in the least squares sense (Equation (5.16)), with no regularization constraint applied.

The actual rotation angles were solved from equation (5.21). This large scale sparse set of linear equations cannot be solved directly due to the singularity without any regularization. The constrained least squares method which is described in Section 5.3.3 was used with the kernel filter $[-0.5\ \ -0.5\ 2\ \ -0.5\ \ -0.5]$ to form the

Table 5.3: The RMSE values and relative errors of grouped angles

Group number	RMSE of estimated angles	relative error (%)
1	0.0016861	5.31
2	0.0014462	4.56
3	0.0014283	4.50
4	0.0013797	4.35
5	0.0014899	4.70
6	0.0014805	4.67
7	0.0014724	4.64
8	0.0018763	5.91
9	0.0016849	5.31
10	0.0017577	5.54
11	0.0014667	4.62
12	0.0014127	4.45
13	0.0014521	4.58
14	0.0012044	3.80
15	0.0013487	4.25

constraint matrix T. The individual rotation angles between two adjacent camera positions are shown in the top of Fig. 5.12. If the rotation angles are ideally uniformly distributed, the angle between two adjacent camera positions should be $\varphi = 0.0021$ rads, or $(\frac{2\pi}{2967})$ rads. The horizontal line is the ideal angle, or φ (rad). The RMSE of estimated angles is 0.00011739 rads, or 5.54% in relative error.

Graphs of the cumulative angles estimated by the proposed method and the cumulative angles assuming that the camera positions are uniformly distributed are not shown because the two lines are too close to each other to be distinguished. A similar result can be obtained by using the total least squares method to solve equation (5.21), as shown in the bottom of Fig. 5.12. The RMSE of estimated angles is 0.00010992 rads, or 5.19% in relative error. We see that both methods give good results.

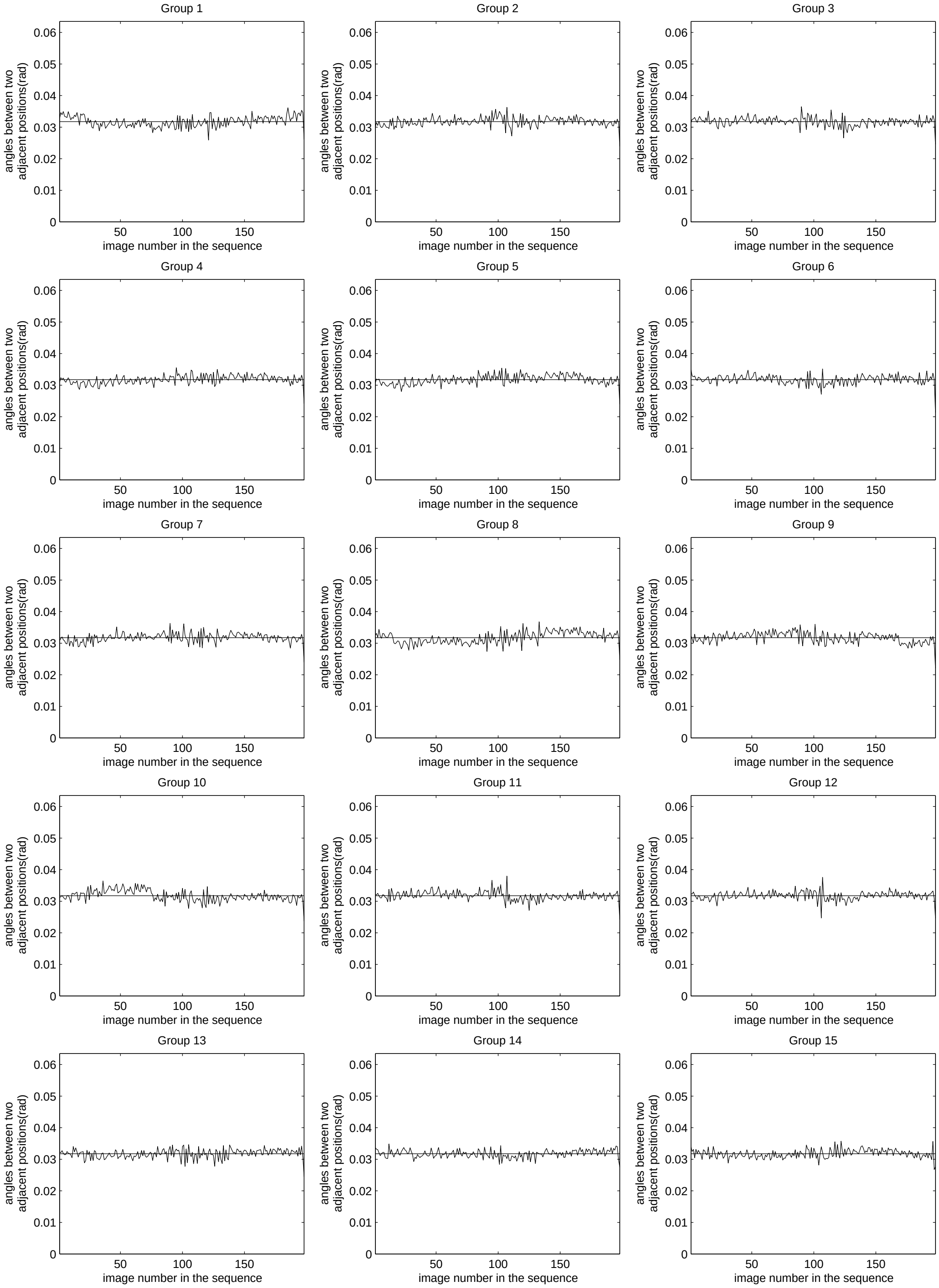


Figure 5.11: The distributions of estimated angles in different groups

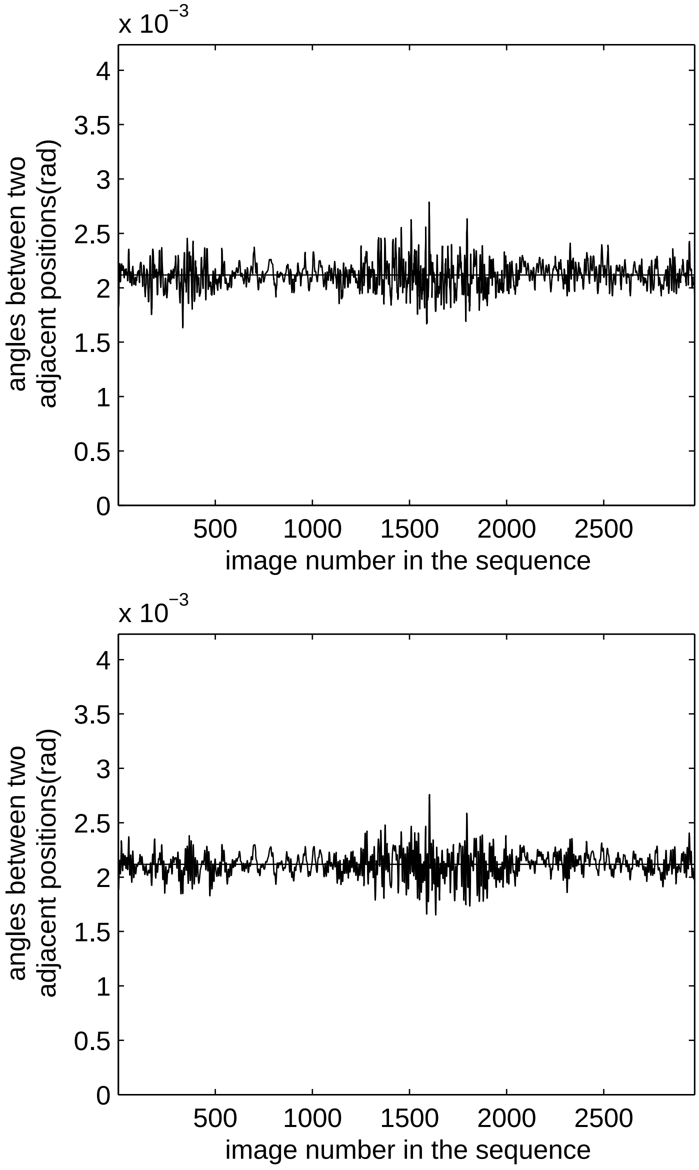


Figure 5.12: The estimated angular positions in simulation II: estimated by constrained least square method (top) and estimated by total least square method(bottom)

5.6.3 Simulation III: rendering with SCOM through non-uniformly distributed pre-captured images

In the last simulation, we tested the proposed rendering algorithm with non-uniformly distributed pre-captured images. The pre-captured images were captured by a CCD web-camera with selected resolution 320x240. The camera was mounted on one end of a long bar, with the other end of the long bar bolted to a desk at its center. The long bar was about 1.5 meters in length and could be manually rotated around the center of the desk. The camera was forced to rotate on a levelled circle because the long bar was held firmly against the desk surface. There were 2309 images taken during one loop. The cumulative angles for the estimated camera positions are shown in Fig. 5.13.

The navigation area is a circular area with a radius of 0.5 meter according to our simulation setup. One of the novel views within the navigation area generated by the proposed rendering algorithm under the constant depth assumption is shown in Fig. 5.14. The rendered image was taken by a virtual camera located at position (0.18m, 45°) in a polar coordinate system, with the polar axis starting from the rotation center and pointing to the position where the first image was taken. The viewing direction is along the 120° direction with respect to the radial direction. We can find that the rendered image is of good quality.

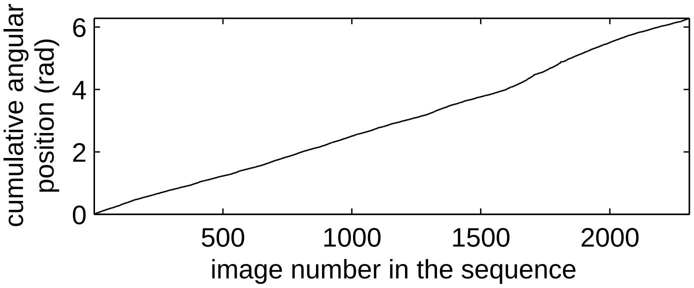


Figure 5.13: Results for simulation III: cumulative angular positions (rad)

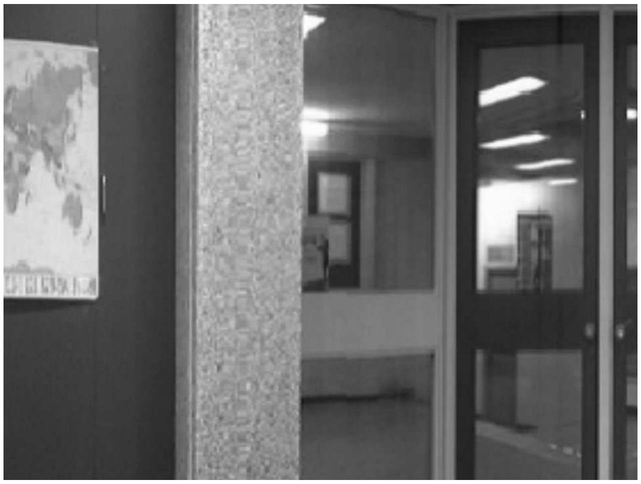


Figure 5.14: Results for simulation III: one of the synthesized views

5.7 Chapter summary

In this chapter, a method to simplify the implementation of the Concentric Mosaics technique has been proposed. This SCOM technique is oriented for an ordinary user to capture Concentric Mosaics data and put it into the Concentric Mosaics rendering framework.

The key issue is to estimate the camera positions, which are on a circle, from the pre-captured images. This strong constraint on camera movement largely simplifies the position estimation compared with the general camera-position estimation problems in computer vision. A stereo-based camera position estimation algorithm is proposed through the solution of a large scale sparse set of linear equations. The coefficients in these linear equations may be noisy due to the estimation errors in the disparity between the matching features and this can possibly make the solution unstable. With a set of associated techniques such as the closed-loop constraint, the method of using ratio-fitting to select good matching features, and the angle grouping technique to reduce the influence from the disparity estimation errors, the effect of the noise in the coefficients can be largely reduced. Moreover, optimization methods, such as applying regularization and using the total-least-squares method, have been proposed to obtain more stable solutions.

The additional error sources that may lower the quality of the synthesized views other than the camera position estimation errors have been illustrated and the correspondent strategies proposed, such as over-sampling the scene and pre-processing the captured images. The synthesized novel views based on the proposed SCOM are of good quality.

Chapter 6

Conclusions and future work

IBR is a relatively new research topic, which requires the combination of the techniques from image processing, computer vision, computer graphics, etc. The fundamental question is how to collect and organize the pre-captured images that are sufficient to represent an environment. Thus, one of the most important issues in IBR is view synthesis, or generating arbitrary novel views in a certain navigation area using the pre-captured images. This thesis focused on various view synthesis methods which could be used for IBR applications. These methods include view mosaicking, view interpolation from adjacent views and view synthesis based on a specific IBR system, Concentric Mosaics.

6.1 Thesis summary

The panoramic view is the simplest method of scene representation for IBR and has been widely used on the Internet. Compared to other techniques for IBR applications, the panorama technique has its own advantages that make it popular and successful. The technical requirements to obtain the pre-captured images are relatively simple and the methods are easy to implement for ordinary users. The quantity of pre-captured image data is relatively small, and can be handled by an ordinary computer. In addition, the methods to generate panoramas are straightforward and many commercial software packages are available.

However, the problems of generating panoramas **of high quality** and **with high resolution** are still open due to the non-ideal motions of the camera during the procedure of taking the pre-captured images. Most current software packages and algorithms to generate panoramas work well for low resolution pre-captured images. In this thesis, the algorithm of generating cylindrical panoramas of good quality using high resolution pre-captured images is studied. The possible registration errors are analyzed and a novel algorithm has been proposed, implemented and tested.

The pre-captured images are taken by a camera mounted on a tripod and rotating around its projection center and the only required calibration parameter of the camera is the camera's focal length. The camera motions that deviate from the ideal ones bring registration errors before stitching. In order to reduce these registration errors, an optimization model is proposed. Based on this model, the methods consisting of a non-linear focal length adjustment and affine transformation are jointly used to minimize the registration errors between two adjacent images. In generating a panoramic view, the registration errors between adjacent images have to be reduced. This usually causes difficulties when registering the first and last pre-captured images at the final step due to lack of the transformation freedom that could be applied. Thus, an algorithm to resolve the above issue is proposed by developing a general coordinate-system-conversion framework. The studies have shown that the proposed algorithms can give good results, compared with some other methods especially in the indoor scenario where the depth variations of the scene are usually quite large.

A large number of panoramic views, or panoramic sequences, is required to build an IBR system with more navigation freedom. However, it is very expensive with current techniques to obtain the required number of panoramas in order to build such an IBR system. Thus, an IBR system with more flexible navigation capability and using fewer pre-captured images is possible if combining the techniques of panoramic view representation and view interpolation.

View interpolation on ordinary planar images is still an open research topic due to the difficulties of obtaining disparity fields between multi-views with similar imaging directions. Moreover, the view interpolation methods in the literature aim at interpolating an intermediate frame in the view sequence or an intermediate view between two images of a pair of stereoscopic images, due to the nature of application requirements. More general camera motion scenarios, including not only translation and rotation but also backward and forward movement, have to be studied due to new IBR application requirements. In this thesis, a new matching-feature-based view interpolation algorithm is first proposed. The algorithm provides a new approach to estimate the disparities between these multi-views with similar imaging directions. Matching features are used in this approach and the simulation results show that this method can interpolate intermediate views of good quality. Then view interpolation when the camera is moving forward and backward is studied. To the author's knowledge, no view interpolation work, addressing such special camera motion scenarios, has been reported in the literature. By upgrading the optical-flow-based disparity estimation approach, a novel disparity estimation and view interpolation algorithm has been proposed and the simulations show that the method can obtain good results.

Light-field-based methods are the recently proposed approaches for IBR. In such approaches, representative light ray sets of the scenes are obtained through the pre-captured images and stored in the database in a specially organized way. The novel views can then be synthesized by combining the light rays from such database.

Special equipment is usually required to build the above databases from the pre-captured images. The cameras that are used to take the pre-captured image are controlled to move along the pre-defined trajectories. Light Field Rendering and Concentric Mosaics are two examples within this category.

However, the technical requirements are too expensive for the ordinary users and this may prevent such approaches from being widely used. A new method, namely Simplified Concentric Mosaics technique, is proposed in this thesis. In the proposed method, the technical requirements for taking pre-captured images are largely reduced compared to the conventional Concentric Mosaics technique. Instead of requiring precisely-controlled mechanical parts, the rotation of the long bar during pre-captured image acquisition in SCOM can be carried out manually. This greatly reduces the application requirements. The proposed novel algorithms to compute camera positions where the pre-captured images are taken require no camera calibration information as opposed to other methods to calculate camera positions in the literature. The algorithms take advantage of the specific camera motion in the SCOM setup to simplify the camera-position-estimation model, which provides efficient solutions for camera position estimation.

6.2 Thesis contributions

The contributions of this thesis are identified below:

- A novel method to stitch two adjacent images with an overlapping area, which are taken by a camera mounted on a tripod and rotated around the camera center, was developed. A model based on affine adjustment together with focal-length adjustment was proposed in this method. This work was published and presented at the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2004 [79]. The application of this algorithm in virtual environments was published and presented at the IEEE International Workshop on Haptic, Audio and Visual Environments and their Applications (HAVE) 2003 [120]. An algorithm based on matching features in the overlap area was implemented.

- In many algorithms for IBR applications, various transformations are involved that need to re-sample the original images. These transformations may include affine transformation, perspective transformation, warping, texture mapping, etc. Different types of transformations sometimes need to be applied on the same images. In this thesis, these transformations are formulated as sampling structure changes that can possibly be cascaded together to minimize the number of interpolation steps and thus minimize the low-pass effect from interpolation procedures. The idea was illustrated in the proposed view-mosaics algorithm.

- When generating panoramas, a set of images that cover a 360° view have to be stitched one by one. These images are usually taken by a camera mounted on a tripod and rotated around its projection center. The accumulated registration errors in the overlap area between the first image and the last image are usually very large when the adjacent images are registered from the first image to the last image. In this case, a complex global optimization algorithm is required to solve the problem. Thus, a novel algorithm was developed and implemented to approach this global minimization solution iteratively without solving the global optimization problem.

- A paper describing a complete study based on the above techniques (for view mosaicking and panorama generation) with the experimental results has been submitted to the IEEE Transactions on Circuits and Systems for Video Technology (CSVT) and is under revision.

- A novel method for view interpolation through triangulation has been developed. It is based on sparse matching features instead of dense disparities and in the scenario that novel views are interpolated from multiple source images (three reference images were used for the experiments). This work was published and presented at the IEEE ICASSP 2005 [121]. Similar work on view interpolation through triangulation with an optimization method to reduce the discontinuities between adjacent triangular patches was published and presented at the SPIE conference Image and Video Communications and Processing (IVCP) 2005 [122].

- A novel algorithm for view interpolation, which addresses the special scenarios that the camera is moving forward and backward, has been proposed. The novel work comes from the new requirements for IBR applications, which was not the case for the view interpolation algorithms in the literature.

- A Simplified Concentric Mosaics (SCOM) technique has been proposed to significantly reduce the technical requirement in the Concentric Mosaics (COM) technique in order to allow an ordinary user to use COM-based technique. The pre-captured images are non-uniformly distributed but a similar rendering algorithm can still be used.

- A novel camera-position-estimation algorithm was implemented for SCOM applications, which requires no camera-calibration information. The algorithm aims at the special camera motion scenario and thus it is more efficient comparing to any other camera-position-estimation algorithms in the literature for SCOM applications. This work was published and presented at the SPIE conference Visual Communication and Image Processing (VCIP) 2005 [123]. Further studies and results with improved algorithms have been submitted to the IEEE Transactions on Circuits and Systems for Video Technology (CSVT) and is under revision.

6.3 Future work

Image-based rendering is a relatively new research topic brought by multimedia application requirements. The techniques are supported by the increasing computing power and data communication capabilities. Although the current research topics are mainly in developing concrete and well-designed methods for various application requirements, the fundamental problem is how to generally collect sufficient pre-captured images to represent an environment.

The basic element for scene sampling and reconstruction is the light ray, according to the theory of plenoptic function representation of the scene. A general theory of scene sampling and reconstruction is one objective for future research. This may possibly only be determined after developing particular concrete techniques, which define various ways of sampling the scenes for arbitrary view reconstruction (synthesis). This thesis is intended to make a contribution in the above directions. The research goal is oriented to the methods which can provide low-cost implementations and generate synthesized views with high quality.

The view interpolation theory and methods developed in chapter 4 and the approaches based on light-field description in chapter 5 are very different techniques. They may be used for different application requirements. The future work following this thesis may include:

1) Further theory and methods for view interpolation from pre-captured images can be studied. By using the techniques from computer vision, the camera positions and shooting directions can be retrieved from the pre-captured images (which can possibly be video sequences). Together with camera calibration information, the 3D structure of the scene can be partially reconstructed if the perfect and full reconstruction is impossible or too expensive. With the obtained 3D model of the scene, the pre-captured image database may be organized in a way that view-interpolation can be implemented more efficiently.

2) The theory and methods for view interpolation studied in this thesis are within the scenarios that the camera positions, from where the pre-captured source images are taken, are close to each other. In the pre-captured image database, many pre-captured images may possibly be taken by the camera at different positions (the distances between these camera positions might be large) but in similar imaging directions (or the imaging directions of the camera towards similar areas of the scene). These pre-captured images are the images of similar scene areas, which are captured at different resolutions, and they could be potentially used jointly to generate one particular view. Thus, view interpolation through the pre-captured images of multiple resolutions is a very important and interesting research topic for future work.

3) Concentric Mosaics is one technique within the category of light-field description. The approaches in this category are promising research directions because they

can almost be scene-independent. This is a good property due to the potential requirement on the hardware/software which can be standardized in the future. In this way, the methods will be easily adapted in the corresponding industries and then will possibly be widely used. The idea of reducing technical requirements for the methods within this category can be extended, such as a Light-Field-rendering technique where the camera is moved in a 2D plane to obtain the pre-captured images.

Bibliography

[1] S. E. Chen and L. Williams, "View interpolation for image synthesis," *Computer Graphics (SIGGRAPH)*, pp. 279–288, August 1993.

[2] S. E. Chen, "Quicktime VR — An image-based approach to virtual environment navigation," *Computer Graphics (SIGGRAPH)*, pp. 29–38, August 1995.

[3] L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system," *Computer Graphics (SIGGRAPH)*, pp. 39–46, August 1995.

[4] M. Levoy and P. Hanrahan, "Light field rendering," *Computer Graphics (SIGGRAPH)*, pp. 31–42, August 1996.

[5] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The Lumigraph," *Computer Graphics (SIGGRAPH)*, pp. 43–54, August 1996.

[6] H.-Y. Shum, S.-C. Chan, and S. Kang, *Image-Based Rendering.* Springer, 2007.

[7] T. Whitted, "Overview of IBR: Software and hardware issues," *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 1–4, September 2000.

[8] P. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based appraoch," *Computer Graphics (SIGGRAPH)*, pp. 11–20, August 1996.

[9] S. B. Kang, M. Wu, Y. Li, and H.-Y. Shum, “Large environment rendering using plenoptic primitives,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1064–1073, November 2003.

[10] E. H. Adelson and J. R. Bergen, “The plenoptic function and the elements of early vision,” in *Computational Models of Visual Processing*, M. Landy and J. Movshon, Eds.

[11] H.-Y. Shum, S. B. Kang, and S.-C. Chan, “Survey of image-based representations and compression techniques,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1020–1037, November 2003.

[12] C. Zhang and T. Chen, “A survey on image-based rendering - Representation, sampling and compression,” *Signal Process., Image Commun.*, vol. 19, no. 1, pp. 1–28, January 2004.

[13] S. M. Seitz and C. M. Dyer, “View morphing,” *Computer Graphics (SIGGRAPH)*, pp. 21–30, Augest 1996.

[14] S. Laveau and O. D. Faugeras, “3-D scene representation as a collection of images and fundamental matrics,” *Proc. IEEE Int. Conf. Pattern Recognition*, vol. A, pp. 689–691, October 1994.

[15] S. Avidan and A. Shashua, “Novel view synthesis in tensor space,” *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pp. 1034–1040, June 1997.

[16] ——, “Novel view synthesis by cascading trilinear tensors,” *IEEE Trans. on Visualization and Computer Graphics*, vol. 4, no. 4, pp. 293 – 306, 1998.

[17] H.-Y. Shum and L.-W. He, “Rendering with concentric mosaics,” *Computer Graphics (SIGGRAPH)*, pp. 299–306, August 1999.

[18] M. M. Oliveira, G. Bishop, and D. McAllister, “Relief texture mapping,” July 2000, pp. 359–368.

[19] S. Vedula, S. Baker, and T. Kanade, “Image-based spatio-temporal modeling and view interpolation of dynamic events,” *ACM Transaction on Graphics*, vol. 24, no. 2, pp. 240–261, April 2005.

[20] A. Laurentini, “The visual hull concept for silhouette based image understanding,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 2, pp. 150–162, February 1994.

[21] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan, “Image-based visual hulls,” *Computer Graphics (SIGGRAPH)*, pp. 369–374, July 2000.

[22] S. M. Seitz and C. M. Dyer, “Photorealistic scene reconstruction by voxel coloring,” *Intern. J. Comput. Vis.*, vol. 35, no. 2, pp. 151–173, 1999.

[23] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *Intern. J. Comput. Vis.*, vol. 47, no. 1, pp. 7–42, May 2002.

[24] L. McMillan, “An image-based approach to three-dimensional computer graphics,” in *Ph.D Thesis.* Department of Computer Science, University of North Carlina at Chapel Hill, 1997.

[25] H. Schirmacher, W. Heidrich, and H. P. Seidel, “High-quality interactive Lumigraph rendering through warping,” *Proc. Graphics Interface*, pp. 87–94, May 2000.

[26] J. Shade, S. Gortler, L.-W. He, and R. Szeliski, “Layer Depth Images,” *Computer Graphics (SIGGRAPH)*, pp. 231–242, July 1998.

[27] C. Chang, G. Bishop, and A. Lastra, “LDI tree: A hierarchical representation for image-based rendering,” *Computer Graphics (SIGGRAPH)*, pp. 291–298, August 1999.

[28] P. Rademacher, “View-dependent geometry,” *Computer Graphics (SIGGRAPH)*, pp. 439–446, Augest 1999.

[29] P. Debevec, Y. Yu, and G. Borshukov, "Efficient view-dependent image-based rendering with projective texture-mapping," *Proc. 9th Eurographics Workshop on Rendering*, pp. 105–116, 1998.

[30] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured Lumigraph rendering," *Computer Graphics (SIGGRAPH)*, pp. 425–432, August 2001.

[31] M. Uyttendaele, A. Criminisi, S. B. Kang, S. Winder, R. Hartley, and R. Szeliski, "High-quality image-based interactive exploration of real-world environment," *IEEE Computer Graphics and Applications*, vol. 24, no. 3, pp. 52–63, 2004.

[32] J. Foote and D. Kimber, "FlyCam: Practical panoramic video and automatic camera control," *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 3, pp. 1419–1422, July 2000.

[33] U. Neumann, T. Pintaric, and A. Rizzo, "Immersive panoramic video," *Proceedings of the eighth ACM international conference on Multimedia*, pp. 493–494, 2000.

[34] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision (2nd Edition)." Cambridge University Press, 2004.

[35] S. M. Seitz and C. M. Dyer, "Physically-valid view synthesis by image interpolation," *Proc. Workshop on Representations of Visual Scenes*, pp. 18–25, 1995.

[36] X. Sun and E. Dubois, "A method for the synthesis of intermediate views in image-based rendering using image rectification," *Proc. 2002 IEEE Canadian Conference on Electrical and Computer Engineering*, vol. 1, pp. 991–994, May 2002.

[37] Y. Xiong and K. Turkowski, "Creating image-based VR using a self-calibrating fisheye lens," *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pp. 237–243, June 1997.

[38] [Online]. Available: http://www.ptgrey.com/prodhucts/ladybug/index.html

[39] R. Szeliski, "Video mosaics for virtual environments," *IEEE Computer Graphics and Application*, vol. 16, no. 2, pp. 22–30, March 1996.

[40] H.-Y. Shum and R. Szeliski, "Construction and refinement of panoramic mosaics with global and local alignment," *Proc. IEEE Int. Conf. Computer Vision*, pp. 953–958, January 1998.

[41] M. Irani, P. Anandan, and S. Hsu, "Mosaic based representations of video sequences and their applications," *Proc. IEEE Int. Conf. Computer Vision*, pp. 605–611, June 1995.

[42] Z. Zhu, G. Xu, E. M. Riseman, and R. Hanson, "Fast generation of dynamic and multi-resolution 360-degree panorama from video sequence," *Proc. IEEE Int. Conf. on Multimedia Computing and Systems*, vol. 1, pp. 400–406, 1999.

[43] M. Uyttendaele, A. Eden, and R. Szeliski, "Eliminating ghosting and exposure artifacts in image mosaics," *Proc. IEEE Conf. Computer Vision Pattern Recognition*, vol. 2, pp. 509–516, December 2001.

[44] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet, "Mosaicing on adaptive manifolds," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 10, pp. 1144–1154, October 2000.

[45] I. Ihm, S. Park, and R. Lee, "Rendering of spherical light fields," *Pacific Graphics*, pp. 59–68, October 1997.

[46] E. Camahort, A. Lerios, and D. Fussell, "Uniformly sampled light fields," *9th Eurographics Rendering Workshop*, pp. 117–130, June-July 1998.

[47] P. P. Sloan, M. F. Cohen, and S. J. Gortler, “Time critical Lumigraph rendering,” *Symposium on Interactive 3D Graphics*, pp. 17–23, April 1997.

[48] D. G. Aliaga and I. Carlbom, “Plenoptic stitching: A scalable method for reconstructing 3D interactive walkthroughs,” *Computer Graphics (SIGGRAPH)*, pp. 443–450, August 2001.

[49] M. Bass, Ed., *Handbook of Optics.* McGraw-Hill, 1995.

[50] H. Aly and E. Dubois, “Specification of the observation model for regularized image up-sampling,” *IEEE Trans. Image Process.*, vol. 14, pp. 567–576, 2005.

[51] X. Sun and E. Dubois, “Scene sampling for the concentric mosaics technique,” *Proc. IEEE Int. Conf. Image Processing*, vol. I, pp. 465–468, September 2002.

[52] A. Katayama, K. Tanaka, T. Oshino, H. Tamura, and S. Fisher, “A viewpoint dependent stereoscopic display using interpolation of multi-viewpoint images,” *Proc. SPIE, Stereoscopic Display and Virtual Reality Systems II*, vol. 2409, pp. 11–20, 1995.

[53] S. Peleg and M. Ben-Ezra, “Stereo panorama with a single camera,” *Proc. IEEE Conf. Computer Vision Pattern Recognition*, vol. 1, pp. 395–401, June 1999.

[54] E. Vincent and R. Laganière, “Detecting and matching feature points,” *Journal of Visual Communication and Image Representation*, vol. 16, no. 1, pp. 38–54, 2005.

[55] H. Moravec, “Towards automatic visual obstacle avoidance,” *Proc. Int. Joint Conf. Artificial Intell.*, pp. 584–590, August 1997.

[56] P. Beaudet, “Rotational invariant image operators,” *Proc. IEEE Int. Conf. Pattern Recognition*, pp. 579–583, 1978.

[57] L. Kitchen and A. Rosenfeld, “Gray level corner detection,” *Pattern Recognition Letters*, vol. 1, pp. 95–102, December 1982.

[58] R. Deriche and G. Giraudon, "Accurate corner detection: An analytical study," *Proc. IEEE Int. Conf. Computer Vision*, pp. 66–70, 1990.

[59] C. Harris and M. Stephens, "A combined corner and edge detector," *Proceedings of 4th Alvey Vision Conference*, pp. 147–151, August 1988.

[60] J. Noble, "Finding corners," *Image Vis. Comput.*, vol. 6, no. 2, pp. 121–128, 1988.

[61] T. Lindeberg, "Feature detection with automatic scale selection," *Intern. J. Comput. Vis.*, vol. 30, no. 2, pp. 77–116, 1998.

[62] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intern. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[63] T. Lindeberg and J. Garding, "Shape-adapted smoothing in estimation of 3-D shape cues from affine distortions of local 2-D brightness structure," *Image Vis. Comput.*, vol. 15, no. 6, pp. 415–434, 1997.

[64] T. Tuytelaars and L. V. Gool, "Content based image retrieval based on local affinely invariant regions," *Proceedings of 3rd International Conference on Visual Information Systems*, pp. 493–500, 1999.

[65] C. Heipke, "Overview of image matching techniques," *Workshop on the Application of Digital Photogrammetric Workstations*, pp. 173–189, 1996.

[66] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 4, pp. 509–522, 2002.

[67] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," *Proc. IEEE Conf. Computer Vision Pattern Recognition*, vol. I, pp. 506–513, 2004.

[68] R. Szeliski and H.-Y. Shum, “Creating full view panoramic image mosaics and environment maps,” *Computer Graphics (SIGGRAPH)*, pp. 251–258, August 1997.

[69] A. Agarwala, C. Zheng, C. Pal, M. Agrawala, M. Cohen, B. Curless, D. Salesin, and R. Szeliski, “Panoramic video textures,” *Computer Graphics (SIGGRAPH)*, pp. 821–827, August 2005.

[70] J. Baldwin, A. Basu, and H. Zhang, “Panoramic video with predictive windows for telepresence applications,” *Proc. International Conference on Robotics and Automation*, vol. III, pp. 1922–1927, May 1999.

[71] S. K. Nayar, “Catadioptric omnidirectional camera,” *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pp. 482–488, June 1997.

[72] S. Peleg and J. Herman, “Panoramic mosaics by manifold projection,” *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pp. 338–343, June 1997.

[73] C. Geyer and K. Daniilidis, “Omnidirectional video,” *The Visual Computer*, vol. 19, no. 6, pp. 405–416, October 2003.

[74] R. Gupta and R. I. Hartley, “Linear pushbroom cameras,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 9, pp. 963–975, September 1997.

[75] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall, “Mosaicing new views: The crossed-slits projection,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 6, pp. 741–754, June 2003.

[76] A. K. Jain, Ed., *Fundamentals of Digital Image Processing.* Prentice-Hall, 1989.

[77] [Online]. Available: http://www.panoguide.com/

[78] M. Traka and G. Tziritas, “Panoramic view construction,” *Signal Process., Image Commun.*, vol. 18, no. 6, pp. 465–481, July 2003.

[79] X. Sun and E. Dubois, “A novel algorithm to stitch multiple views in image mosaics,” *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. III, pp. 481–484, May 2004.

[80] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach.* Prentice Hall, 2003.

[81] S. Mann and R. W. Picard, “Virtual bellows: Constructing high-quality images from video,” *Proc. IEEE Int. Conf. Image Processing*, vol. I, pp. 363–367, November 1994.

[82] M. Brown and D. G. Lowe, “Automatic panoramic image stitching using invariant features,” *Intern. J. Comput. Vis.*, vol. 74, no. 1, pp. 59–73, 2007.

[83] E. Vincent and R. Laganière, “Matching with epipolar gradient features and edge transfer,” *Proc. IEEE Int. Conf. Image Processing*, pp. 277–280, September 2003.

[84] G. Roth. Projective vision toolkit. [Online]. Available: http://www.cv.iit.nrc.ca/ ∼ gerhard/PVT/

[85] A. Whitehead and G.Roth, “The projective vision toolkit,” *Proc. Modelling and Simulation*, pp. 204–209, May 2000.

[86] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing.* Upper Saddle River, NJ: Prentice Hall, 2000.

[87] L. G. Shapiro and G. C. Stockman, *Computer Vision.* Prentice Hall, 2001.

[88] J. Davis, “Mosaics of scenes with moving objects,” *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pp. 354–360, 1998.

[89] D. M. Mount, N. S. Netanyahu, and J. LeMoigne, “Efficient algorithms for robust feature matching,” *Pattern Recognit.*, vol. 32, pp. 17–38, 1999.

[90] S. Baker, T. Sim, and T. Kanade, “When is the shape of a scene unique given its light-field: A fundamental theorem of 3D vision?” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 1, pp. 100 – 109, 2003.

[91] P. S. Heckbert, “Survey of texture mapping,” *IEEE Computer Graphics and Applications*, vol. 11, no. 6, pp. 56–57, 1986.

[92] M. Lhuillier and L. Quan, “Image-based rendering by joint view triangulation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1051–1063, 2003.

[93] Z. Zhang, L. wang, B. Guo, and H.-Y. Shum, “Feature-based light field morphing,” *Computer Graphics (SIGGRAPH)*, pp. 457–464, July 2002.

[94] J. R. Shewchuk, *Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator*, ser. Lecture Notes in Computer Science, M. C. Lin and D. Manocha, Eds. Springer-Verlag, May 1996, vol. 1148, from the First ACM Workshop on Applied Computational Geometry.

[95] J. Gomes, B. Costa, L. Darsa, and L. Velho, *Warping and Morphing of Graphics Objects*. Morgan Kaufmann, 1998.

[96] Y. Horry, K. Anjyo, and K. Arai, “Tour into the picture: Using a spidery mesh interface to make animation from a single image,” *Computer Graphics (SIGGRAPH)*, pp. 225–232, August 1997.

[97] Model house image sequence. [Online]. Available: http://www.robots.ox.ac.uk/ vgg/data/

[98] O. Faugeras, L. Robert, S. Laveau, G. Csurka, C. Zeller, C. Gauclin, and I. Zoghlami, “3-D reconstruction of urban scenes from image sequences,” *Coputer Vision and Image Understanding*, vol. 69, no. 3, pp. 292–309, 1998.

[99] X. Huang and E. Dubois, “3D reconstruction based on a hybrid disparity estimation algorithm,” *Proc. IEEE Int. Conf. Image Processing*, pp. 1025–1028, October 2006.

[100] ——, "Disparity estimation for the intermediate view interpolation of stereoscopic images," *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, pp. pp. II–881–II–884, March 2005.

[101] L. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. 29, pp. 1799–1808, 1981.

[102] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. 7th International Conference on Artificial Intelligence*, pp. 121–130, 1981.

[103] S. Ince and J. Konrad, "Recovery of a missing color component in stereo images (or helping NASA find little green martians)," *Proc. SPIE Image and Video Communications and Processing*, pp. 127–138, January 2005.

[104] J. Konrad and E. Dubois, "Bayesian estimation of motion vector fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 910–927, 1992.

[105] E. Dubois and J. Konrad, "Estimation of 2-d motion fields from image sequences with application to motion-compensated processing," in *Motion Analysis and Image Sequence Processing*, M. Sezan and R. Lagendijk, Eds. Kluwer Academic Publishers, 1993, ch. 3, pp. 53–87.

[106] M. Bierling, "Displacement estimation by hierarchical block matching," *Proc. SPIE Visual Communications and Image Process.*, vol. 1001, pp. 942–951, 1988.

[107] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards.* Kluwer Academic Press, 1997.

[108] J.-X. Chai, S. B. Kang, and H.-Y. Shum, "Rendering with non-uniform approximate concentric mosaics," *2nd Workshop on 3D Structure from Multiple Images of Large-scale Environments*, pp. 94–107, 2000.

[109] G. Jiang, Y. Wei, L. Quan, H. Tsui, and H.-Y. Shum, "Outward-looking circular motion analysis of large image sequences," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 2, pp. 271–277, 2005.

[110] R. Szeliski. (2001) Lecture notes (winter, 2001): Vision for Graphics (University of Washington). [Online]. Available: http://www.cs.washington.edu/education/courses/cse590ss/01wi/

[111] G. Roth and A. Whitehead, "Using projective vision to find camera positions in an image sequence," *Proc. Conf. Vision Interface*, pp. 225–232, May 2000.

[112] E. Izquierdo and J.-R. Ohm, "Image-based rendering and 3D modeling: A complete framework," *Signal Process., Image Commun.*, vol. 15, pp. 817–858, 2000.

[113] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[114] A. Gersho and R. M. Gray, *Vector quantization and signal compression.* Kluwer Academic Press, 1992.

[115] B. R. Hunt, "The application of constrained least squares estimations to image restoration by digital computer," *IEEE Trans. Comput.*, vol. 22, no. 9, pp. 805–812, 1973.

[116] G. H. Golub and C. F. Van Loan, "An analysis of the total least square problem," *SIAM Journal on Numerical Analysis*, vol. 17, pp. 883–893, 1980.

[117] Z. F. Gan, S.-C. Chan, K. T. Ng, and H.-Y. Shum, "On the rendering and post-processing of simplified dynamic light fields with depth information," *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 3, pp. 321–324, May 2004.

[118] M. Wu, H. Sun, and H.-Y. Shum, “Real-time stereo rendering of concentric mosaics with linear interpolation,” *Proc. SPIE Visual Communications and Image Process.*, vol. 4067, pp. 23–30, 2000.

[119] A. Whitehead. Projective vision toolkit. [Online]. Available: http://iv.csit.carleton.ca/ awhitehe/PVT/

[120] X. Sun and E. Dubois, “Augmented reality: a novel approach for navigating in panorama-based virtual environments,” *Proc. HAVE 2003: IEEE International Workshop on Haptic, Audio and Visual Environments and Their Applications*, pp. 13–18, September 2003.

[121] ——, “A matching-based view interpolation scheme,” *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. III, pp. 877–880, March 2005.

[122] ——, “View morphing and interpolation through triangulation,” *Proc. SPIE Image and Video Communications and Processing*, pp. 513–521, January 2005.

[123] ——, “A simplified concentric mosaics system with non-uniformly distributed pre-captured images,” *Proc. SPIE Visual Communications and Image Processing*, vol. 5960, pp. 1255–1266, July 2005.