Building Panoramas from Photographs Taken with a Hand-held Camera

by

Chen, Hui
陳　輝

A dissertation submitted to
the University of Hong Kong
for the degree of
Doctor of Philosophy

January 2002

Abstract of dissertation entitled

" Building Panoramas from Photographs Taken with a Hand-held Camera "

submitted by **Chen, Hui**

for the degree of **Doctor of Philosophy**

at the University of Hong Kong

in **January, 2002**

This thesis presents a method for building a full view cylindrical panorama from uncalibrated photographs taken naturally with an ordinary hand-held camera held at an approximately fixed location.

Such photographs usually have large perspective distortions, a small amount of overlap, possible brightness differences, and unintended camera rotations(tilt and roll). These characteristics make both image registration and panorama building more difficult than when using photographs taken by cameras calibrated by special equipment.

We show how these images can be registered by a feature matching scheme. The features used are based on local edge gradient intensities and shape matching. Preprocessing is used to quickly reject impossible matches. When feature correspondence is inadequate by itself, we further invoke an optical flow based fine registration process to improve the registration.

We show how to composite these photographs into a panorama with perspective transformations between adjacent images; these perspective transformations are determined by minimizing errors between corresponding features from adjacent images, using a linear method. Further corrections are then made to the panorama to allow for the fact that a local pairwise image registration method is used, rather than a global solution.

Experiments show that our method yields visually satisfactory results from real photographs taken with modest care to ensure a reasonable amount of overlap.

Department of Computer Science and Information System
University of Hong Kong

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Constructing a complete panorama of a 3D scene from a sequence of partially overlapping photographs is one of the fundamental modeling tasks in building an image-based virtual reality system (IBVR), and also has other applications for visualization. In recent years, image-based virtual reality systems have been a focus of interest for both computer vision and computer graphics communities. Compared with conventional 3D-model based VR systems, IBVR systems have the advantage of photographic realism and rendering simplicity. The most popular IBVR systems are based on nodal panoramas [Chen93, Chen95, McMillan95, Szeliski96, Kang98, Shum00], in which a full-view panorama is constructed from several panorama nodes and in-between views are generated by interpolation of these nodes. Others construct panoramas with a moving viewing center [Peleg00, Rademacher98], or by dense sampling of the environment using light fields and lumigraphs [Levo96, Gortlet96]. Being an effective tool to present a photo-realistic virtual reality with moderate storage, nodal panorama images are very popular on the World Wide Web and in multimedia applications. However, creating high quality panoramas, especially those that form a closed shell, remains difficult.

With the rapid growth of computational power of personal computers, it is envisioned that the high demand for computer graphics and virtual reality applications, even by home computer users, will increase significantly. Thus, reliable automatic construction of panoramas from a series of photographs taken with a hand-held camera is a topic of interest.

In order to build a full-view panorama, first we need to acquire a set of images, which can be done either with an ordinary camera or by using special equipment. Since a *single* ordinary image can usually only capture a small portion of the environment, a sequence of overlapping images must be taken to cover it completely. A common coordinate system for all images must be determined for stitching the images together to composite a panorama. Determining the transformation between the local coordinate systems of two images is a crucial problem. Finally, the panorama is mapped onto a surface model for users to view. We study two major sub-problems in constructing a panorama: image alignment and panorama composition. A prototype of the system has been built using a Pentium III 500MHz PC to validate the theoretical ideas presented here.

In this chapter, I first present the background of our research in Section 1.1, and then give a detailed description of the problem in Section 1.2. In Sections 1.3, I review relevant previous work. The contributions and an outline of the thesis are given in Sections 1.4 and 1.5.

1.1 Background

A panorama is an image with a large field of view. Generally, in computer graphics a panorama is taken to be a compact representation of some environment viewed from either a single node or along a moving path, and the panorama captures either

the whole or a part of the environment. For single node panoramas, there is usually a full 360° field of view in a horizontal or a vertical direction, or in both. Previous work such as [Chen95, Szeliski97, Shum00, Bao99, Xiong98] studies how to build this kind of panorama. Moving path panoramas in the simplest case provide an orthogonally viewed panoramic image of a large scene, as described by [Rousso97]. More generally, a manifold projection is performed: see [Peleg97, Peleg00], for example. The basic steps in building a panorama are image acquisition, image alignment and panorama composition. In the rest of this section, we will briefly examine these issues.

1.1.1 Image Acquisition

To build a panorama, we first need to acquire input images. Input images can be obtained by different methods. Depending on the choice of method, the subsequent processing of these images will be different. Images can be captured from a single viewing center using two techniques, *large view imaging*, using special equipment and *ordinary perspective imaging*, using conventional cameras to take a series of images in different viewing directions. Images captured from a moving viewing center, but with fixed viewing direction, use methods of *multiperspective imaging*. The problem studied in this thesis concerns ordinary perspective imaging, but we first review each of these approaches.

Large View Imaging

Images with a large field of view can be directly recorded by special equipment, such as panoramic cameras and omnidirectional cameras, methods referred to as *large view imaging*. A panoramic camera can directly capture a cylindrical panoramic

image by recording an image onto a long film-strip [Meehan90]. An omnidirectional camera can capture a hemispherical field of view using mirrored pyramids and parabolic mirrors [Nayar97, Onoe98].

Another type of camera which can capture a nearly hemispherical field of view uses a fish eye lens camera[Xiong97]. This approach, however, requires a special calibration algorithm to construct distortion-free perspective images of the viewed scene.

Such cameras can typically capture an entire environment with just one, two or four images. In the latter cases, the small number of images to be registered can help to keep computational time low. However, the limited resolution of each shot taken using wide-angle cameras compromises the panoramic image resolution. Also, such special equipment is relatively expensive and is not generally available.

Ordinary Perspective Imaging

Images can also be captured using readily available equipment, such as ordinary cameras, digital cameras and video cameras. We refer to the acquisition of images by such methods from a fixed viewpoint as *ordinary perspective imaging*, or *ordinary imaging* for short. Images taken in such a way are related to each other by perspective transforms. Ordinary imaging pictures usually have a smaller field of view than our human view. To build a panoramic image with a larger field of view, we need to take a sequence of adjacent or partially overlapped images, determine the transformations between neighboring images, and stitch all the images together. We refer to such a set of images as an *image mosaic*. Finding the space transforms between the images and stitching the images together is referred to as *image mosaicking* [Irani96, Irani98a].

Simple image mosaics can be created by rotating the camera around its optical

center using a special device, such as a turntable that provides accurately known transformations between the images [Chen95]. Alternatively, the images can be obtained by a hand-held camera as long as they are captured from approximately the same viewing position. In our research, we are aimed to simplify input image acquisition, and to build a panorama from images taken with an uncalibrated hand-held camera.

Multiperspective Imaging

Another type of image with a large field of view is captured by the *multiperspective imaging* method, where image strips to be composited are used to record images taken from a smoothly moving viewing center. One way to capture the strips is to use a one-dimensional cameras—a strip camera [Ghosh88] or a push-room camera [Hartley94] for scanning the environment. The other is to cut strips from a sequence of two-dimensional images taken with a conventional camera [Peleg97]. This sequence of strips can directly acquire orthographic maps with translational motion, images along an arbitrary path, and multi-center-projection images [Peleg97, Zheng92, Rademacher98, Wood97]. Methods for processing such a sequence of strips into a panorama can handle a wide variety of viewing motions including motion towards the scene and optical zoom [Zheng99, Peleg00].

Such methods, however, suffer from several disadvantages. The fact that the viewing center is moving means that, in the general case, some approximations must be made—for example, that the depth differences in the scene are negligible. Also the scene is not uniformly sampled when the viewing direction is not vertical with the camera motion direction, leading to poor quality of the final panorama, which is exacerbated by the multiple warpings required.

1.1.2 Image Alignment

When using an ordinary hand-held camera, we need to stitch together overlapped views of a scene to form a larger field of view. To be able to do this, the transformation between the photographs must be determined. This is usually done by finding transformations between adjacent images. This is a problem of image alignment or *registration*. Image registration can be done either manually or automatically. Manual methods involve interactively translating and rotating the second image of each pair into its correct relative position, or interactively identifying correspondences in a pair of images and using them to compute a space transformation. Automatic image registration has been studied for decades and is a central issue of many research areas. We briefly review this subject in the following.

Overview

Image registration problems arise when images of the same scene are taken from different viewpoints or viewing directions, possibly with different sensors, different lighting conditions, and at different times. Some examples of the need for registration are: stitching satellite images, matching biomedical images for diagnosis, matching stereo images for reconstructing depth or shape, and matching objects for recognition. To bring multiple images into alignment is one of the most extensively studied problems in areas including computer vision, pattern recognition, medical image analysis, and remotely sensed data processing. Numerous practical and theoretical studies have been performed on this subject for several decades.

The relationship between a pair of images could be a single global transformation, if for example the images are taken from different views of a static planar surface, or it can be a possibly discontinuous spatially varying transformation, as

might arise in images taken at different times of scenes containing moving objects. In either case, determining this relationship (the registration problem) is a difficult one and often complicated by as occlusion, ambiguous matches, and the presence of observation noise or distortion. We are interested in finding a single global transformation in our study, where a single equation maps the entire image, since we assume the viewing center is roughly fixed.

The goal of image registration is to determine the spatial transform that matches pixels in one image to similar pixels in another. The problem of registration can be eased by constraining the type of transformation field to be estimated. The simplest case is to assume that the field is translational, which is sometimes a reasonable assumption in tasks such as motion or depth estimation [Chiang93, Dhond89, Aschwanden93]. A more general approach is to assume that the images are related by a six parameter affine transformation, corresponding to dilation, rotation, shear and translation. For example, such a model is appropriate in the case of a planar surface viewed under the assumption of weak perspective projection [Li95, Zheng93]. The affine approach gives considerable flexibility in estimating a wide range of transformation fields and has been adopted in a large number of registration techniques. A more general model is an 8 parameter perspective transformation, which arises for a planar surface under translation as well as 3D rotation, or a camera under 3D rotation about a fixed viewpoint [Bao99, Chen00]. When a transformataion does not belong to any of the above models, a polynomial transformation can be used to approximate the distortions between two images as long as the distortion is not too great, such as distortions due to moderate terrain relief [Brown92, Singh92]. A good approximation needs higher order terms, and thus more parameters are required.

The transformation parameters can be obtained by minimizing either a sum of

squared intensity difference at corresponding pixels or a sum of squared Euclidean distance of corresponding pixels. These minimization problems are linear in the cases of 2-parameter translation, 6-parameter affine transformation, and multi-parameter polynomial transformations, but are non-linear in the case of perspective transformation. To perform the optimization, the polynomial approach needs a large number of correspondences since it generally contains more parameters; hence, it is computationally expensive. The non-linear minimization problem involved in perspective transformation can be solved by a Gauss-Newton method or a Levenberg-Marquardt method [Press92, Thevenaz98]. However it suffers from sensitivity to local minima and high computational expense.

Methods

Generally, there are two basic classes of approach to image registration, non-feature based and feature based [Pratt91, Haralick93, Deriche93].

The first class comprises non-feature based methods that minimize the intensity differences between two images, and are referred as *optical flow based* or *gradient-based*, where the spatial and temporal derivatives of image intensity are used to describe image velocity—the optical flow.

Methods described in [Bergen92, Irani95, Sawhney99, Szeliski97, SzelS97, Aubert99 belong to this category. A basic assumption of these methods is that, when two images are perfectly aligned, the intensities of corresponding pixels are matched exactly in the overlap region. Thus, the objective function is a sum of squared intensity differences of corresponding pixels over the whole area. The objective function is minimized via the Gauss-Newton optimization technique. These methods have the advantage that neither feature extraction nor feature matching is required. Another advantage is their high accuracy because they use information

from all pixels in the overlap region—a full density approach. However, the gradient method assumes a small spatial and temporal variation of image intensity in formulating the problem. Due to this locality of the intensity gradient constraint, large pixel displacements cannot be accommodated.

The second class comprises feature based methods [Brown92]. Features (salient points) are first selected in each of the two images, and corresponding features are matched according to some similarity metric. An optimization procedure is then used to compute a transformation that aligns features from one image with corresponding features in the other image. Feature points can be corners, centroids of closed contours, distinctive texture points, or other salient points [Zoghlami97, Harris88]. Two feature points are deemed to correspond if a small window centred at one feature in the first image is similar to a window of the same size centred at the other feature in the second image. This method of feature matching is called *template matching.* The similarity measure can be based on image attributes like intensity distributions, Fourier spectra [Kuglin75, Kruger98], or wavelet coefficients [Bao99, Zheng93]. The measurements can be *normalized cross-correlation (NCC)*, or *sum of squared difference SSD. NCC* correlates two image windows by multiplication of each pixel's squared mean difference of attribute, while *SSD* finds difference of two images by subtraction and is thus computationally more efficient. Though *NCC* is costly to compute, it is generally preferred due to its invariance to linear change of pixel attribute between matching windows. Other matching methods include, non-correlation-like flexible histogram models [Bonet98], structural attributes of extracted objects [Ventura90], contour-chain codes [Li95], surfaces [Maurer98], elastic contour matching [Li95, Davatzikos96], moment invariants [Goshtasby85, Super95] and convex hulls [Yang99]. These feature matching primitives have more disambiguating power compared to intensity values. Among them,

some are better for remote sensing images [Li95, Zheng93, Ventura90, Bonet98], some are more effective on multi-channel medical images [Davatzikos96, Chiang93, Singh92], and others, like Fourior spectra, are well suited for images with frequency-depended noise [Castro87, Reddy96]. Methods in this category can usually register images with large pixel displacements.

A hierarchical fine-coarse strategy is often used in both classes of registration algorithms above to enlarge registration scope, to raise efficiency, or to avoid local minima.

In the first class of gradient-based approaches, the advantages are that they do not need to perform the difficult and computationally intensive tasks of feature extraction and feature matching; and they can achieve high registration resolution due to the use of full density pixel information over the image. The disadvantages are that they require a coarse alignment within a few pixels and that no large intensity changes between the images are allowed. The second class of feature based methods can generally align images with much larger spatial displacement and are more tolerant to brightness difference and other distortions. The shortcoming of these approaches are that they are less accurate when the number of feature correspondence is not sufficiently large and not well distributed in the whole region, or the features are not precisely located, or false correspondences are present. Further, computational cost increases quickly with the growth of the number of correspondences.

Both classes of approach have their merits depending on what kind of images are to be registered and what is already known about the transformation. Over the years, a broad range of techniques has been developed in the subject. However, image registration for panorama construction has several specific requirements which many of the general methods reviewed above do not always address:

- A perspective transformation, not just an affine one, is required to relate the image pairs.

- Large brightness differences may exit between corresponding points in the two images.

- There may be small overlap, and the amount of overlap may not be known before hand—coarse registration may be needed as well as fine registration.

- The method must work robustly for different types of scenes, and scenes of high complexity.

As a result, we have derived special purpose methods for image registration, relying on the principles and general ideas already surveyed; these ideas must be adapted to suit for the panorama construction problem.

1.1.3 Panorama Composition

Given a sequence of images taken from a single viewing position, and the transformations between them, any new view taken in an arbitrary direction from that viewing position can be obtained directly by sampling and blending the contributing images. However, using this method to generate new images is hard to do in real-time. A better approach is to composite a single panoramic image from the source images, projecting each source image to the pixel map of the panoramic model, which is then used to render new views. This can satisfy the real-time requirement. Panoramas can be of several forms, such as rectilinear, cubical, cylindrical or spherical, as will be described below. The cylindrical panorama is the most commonly used. The pixel map of the panoramas is a 2D image, where a reference image is first chosen from the sequence, and all other images are

warped in the 2D coordinate system attached to this image by the relative spatial transformations between them. To composite a panoramic image, the form of the panorama is first chosen, then the source images are warped, sampled and blended. As errors exist in transformation parameter estimation, which are propagated in the registration process, further corrections are required to correct global errors in the final panorama, as described in section 1.3. Next several common forms of panorama are outlined.

Rectilinear Panorama

A rectilinear panoramic image is a set of images warped and clipped to rectangular frames which are placed at equal angles about an axis. This approach is used in [Szeliski96] for stitching a video sequence together to form a wide scene. It has a problem of unequal sampling in that there is a different density of information at the center of each frame compared to its edges.

Cubical Panorama

A cubical panoramic image is a set of six planar projections in the form of a cube viewed from a center point. This approach has been used in [Greene86, Qickvr, VRML] for the representation of a complete scene. Such a representation can easily be stored and accessed systematically. However, it has significant problems relating to its acquisition and registration. Also, computation of image flow fields when using multiple cube maps in IBVR applications is not straightforward. It also has the problem of unequal sampling.

Spherical Panorama

A spherical panorama projects the source images onto a unit sphere surface centered about the viewpoint [Szeliski97, Xiong97]. It is obtained by warping and stitching a source set of images taken two degrees of freedom of viewing direction, and 360° viewing in both directions. The spherical pixel map can be stored in a latitude-longitude format, or in an uniform format with equal area sampling. A spherical panorama is the most natural description of a complete 3D scene, but it lacks a suitable pixel map representation. If it is stored using uniform equal area sampling, it is generally ill-suited for systematic access as a data structure. If it is stored using simple neighborhood relationships such as a latitude-longitude map, the pixels are non-uniform, and generally quite distorted when mapped to a plane. When rendering a new rectangular image, the pixel positions of the panoramic map are generally on a curved grid, thus requiring complex reconstruction and re-sampling.

Cylindrical Panorama

A cylindrical panoramic image is a wide scene image stored in the form of a cylindrical map [Chen95, McMillan95, Szeliski97, Kang98] which is obtained by compositing a set of regular images and warping them onto a cylindrical surface. It provides a limited vertical view. Since the scene on the top and bottom are usually sky, ceiling and ground which are generally not interesting, a cylindrical panorama is good enough for representing the scene in most applications.

Cylindrical panoramas are commonly used also because of their ease of construction, and because it can be easily unrolled into a simple planar map that can be easily stored and accessed by a computer in a systematic way. One shortcoming

is the boundary conditions introduced at the top and bottom. Usually, it is chosen not to put end caps on the map.

Both cylindrical and spherical maps will be discussed further in this thesis. However, we concentrate on cylindrical maps.

1.2 Problem Statement

The main goal of this thesis is to develop techniques which will allow end-users to easily build a cylindrical panorama for image-based VR and other applications from a sequence of uncalibrated photographs taken naturally with a hand-held camera, i.e., without the need for sophisticated calibration devices or turntables. Such photographs will be called *natural photographs*. Such images fall into category of ordinary perspective imaging. The panorama to be built is a single nodal panorama having a 360° degree field of view in one direction. We also consider briefly spherical panoramas.

In our research, we assume that the photographer uses reasonable efforts to keep the camera at a fixed location while taking a series of natural photographs, with some overlap between adjacent views. It is assumed that the camera is only rotated in a horizontal plane (*panned*), and that tilting of the camera towards the ground or sky (*tilting*), and rolling of the camera about an axis pointing into the scene(*rolling*) are kept to a minimum. The focal length of the lens is assumed to be the same for all photographs.

Specifically, our assumptions about the natural photographs from which we will build a panorama are:

- All photographs are taken from an (approximately) fixed view point.

- The photographs provide a 360° view of the scene.
- The same focal length is used for all the photographs in the sequence.
- Any two consecutive photographs should overlap by about 1/6 or more of the width of each photo.
- Camera roll and tilt should be kept to a minimum.

In practice, our method compensates for large panning and tilting. However, large tilting will produce unsatisfactory cylindrical panorama, no matter what method is used.

We believe that these requirements should be fairly easy for an amateur photographer to meet. Examples of images acquired in this way are shown in Figure 1.1. The methods developed in this thesis composite such a sequence of photographs to produce a panorama, like that shown in Figure 1.2.

We view the problem as involving two major sub-problems:

- Image alignment: the projective registration of adjacent overlapping images into the same coordinate system. As long as the images are taken from a nearly fixed viewpoint, their relative transformation can be approximated by a perspective transform. We wish to find robust methods to automatically determine this transform.
- Panorama composition: the source images are projected onto a cylindrical or spherical pixel map. Pixels in each overlap region are blended from multiple images. Various global errors in transformation estimation must be corrected so that the source images represent a closed surface seamlessly.

The difficulties in computing image alignment are perspective distortion, relatively small overlap between adjacent images and brightness differences of the

same area in different images. Reducing the number of images can make the combination process computationally more efficient, so we want a small set of images which cover the environment. However, doing so requires large changes in viewing orientation, producing a greater amount of perspective distortion and potentially relatively little overlap. Also, the possibility of large exposure differences between adjacent images is increased. These considerations make image registration more difficult than in traditional applications like medical imaging and remote photography.

In complete (closed surface) panorama composition where we start from an initial image and work outwards, we also need to overcome the propagated pairwise registrations errors, which result in a mismatch on return to the starting image. We must also correct other errors arising due to incorrect initial estimates of focal length, and tilting and rolling of the initial image.

Here we consider lens distortion to be negligible. Correction of this distortion can be found in literature on camera calibration [Zhang00, Faugeras00, Hartley94-1, Stein95].

1.3 Previous Work

The requirement to visualize a scene with a large field of view has occurred since the beginning of photography, because a camera's field of view is smaller than the human field of view. Earlier works on fusing small images to construct a larger one can be found in [Milgram75, Milgram77, Shiren89, Peleg81, Burt83], with applications to aerial and satellite images.

Image VR applications: the visualization of a full-view scene, or building a full-view panorama, have been actively explored in the past few years. Recent work on

I_0 I_1 I_2 I_3

I_4 I_5 I_6 I_7

Figure 1.1: Image sequence

Figure 1.2: Constructed panorama

constructing a composite panorama from images taken from a common viewpoint with an uncalibrated camera can be found in [Chen95, Szeliski97, Shum00, Bao99]; explicitly determining the camera calibration effect is considered in [Xiong98]. Approaches to constructing panorama from images taken from a moving viewpoint are found in [Peleg97, Rousso97, Peleg00]. [Kang99] has studied the various error effects in building panoramas. There are also commercial panorama building products, which we will also consider.

Quicktime VR

A notable commercial product for image-based VR is Apple's QuickTime VR [Chen95, Qickvr], which stores cylindrical panoramas at discrete nodes (viewing positions), and allows a user to rotate at those nodes to produce a new view of the scene. More recently, these ideas have been extended to also allow cubical panoramas. To build a cylindrical panorama, QuickTime VR provides a tool to perform image stitching, which requires that pictures are taken from a single viewpoint or center of projection by panning the camera around its optical center using a special device. This device can take pictures which are separated by approximately equal horizontal panning angles with minimal tilting and rolling. The software requires the pictures to have about 50% overlap and the camera to have a known focal length. These specifications simplify the image registration problem. Automatic image alignment is performed by a correlation-based method which searches for image correspondences. Our experience of using this software shows that there is still much room left for improvement. Even though the hardware provides a good initial estimate of registration, the software can still fail to register adjacent images.

We have also tried using the Quicktime VR software tools with a handheld camera without a special tripod and rig, when it is not simple to ensure a pure horizontal panning. Small tilting and rolling may exist in each images. Thus, image alignment becomes more difficult, as does the composition of the final panoramic map, where some rectification must be made for generating a seamlessly closed cylindrical panorama. The Quicktime VR software does not make such corrections.

Szeliski and Shum's Approach

Automatic alignment of images separated by small rotations in 3D is studied in [Szeliski97, Shum00]. They present a method to create a panorama from a set of images taken from a fixed viewpoint without camera calibration. Their approach applies a hierarchical gradient based motion estimation framework [Bergen92] for stitching panorama images. Two adjacent images are related using one of three alternative motion models: a 2-parameter translation model, or an 8-parameter perspective transformation model, or a 3-parameter rotation model.

In the first approach, each source image is warped onto a suitable cylindrical surface with a known focal length. A 2-parameter translation model (x and y for unwrapped cylinder) is applied to align the cylindrically mapped images. A gradient-based registration method is used to recover the 2D translation required. However, their warping method is only strictly correct if a pure panning motion has occurred with no tilting and rolling.

In the second and third approaches, they represent image mosaics as a collection of images with associated geometric transformations, rather than projecting all images onto a common surface to composite a panorama. The transformations comprise an 8-parameter perspective transformation model or a 3-parameter rotation model(which does allow for tilting and rolling), which are also found using gradient-based registration methods. This representation can avoid the singularity problem on the top and bottom when projecting the images onto a cylinder surface. The disadvantage of this mosaic representation is the slow rendering speed, as blending must be recomputed for each new image required.

The limitation of their gradient-based alignment approach is that it can only register displacements already known to within a few pixels. They propose using

a hierarchical coarse-fine algorithm to enlarge the region of convergence, but this may still fail when the initial displacement exceeds a certain amount. Thus, when starting with an overlap of less than 50%, their hierarchical algorithm does not work. To overcome this problem, in [Shum00], they suggest using a correlation-style brute-force search on small patches of adjacent images to find a good initial alignment before using the gradient-based registration algorithm. This brute force correlation search is very expensive and not particularly robust. In addition, no steps are taken to exclude false matches.

In compositing a cylindrical panorama, an important task is to close any end mismatch due to various causes. Shum et al [Shum00] consider errors of this sort caused by use of incorrect focal length and propagated misregistration errors over the sequence of images. To deal with these problems, they first suggest a focal length adjustment method to force the panorama ends to meet, which actually stretches or shortens the panorama in width. It is not an ideal solution since it introduces an aspect ratio distortion, in proportion to the focal length error. They also suggest a feature-based global registration method intended to carry out end closing, where they minimize the errors of correspondence in all images simultaneously. However, in their objective function, the starting and end images are not explicitly fixed in such a way as to impose a closing constraint, i.e., that the whole closed claim of transformation is represented by an identity transformation. Thus, they actually perform a multiple registration but *not* a global registration.

To remedy the blurring caused by small camera translations, they suggest a deghosting (local alignment) technique, i.e., to divide the images into smaller patches each of which is registered separately. Together with their block adjustment (global alignment) technique, it improves the quality of panorama image. One problem with this deghosting method is outside the overlapping region there

is nothing to register remaining image patches with: there is no uniform transformation which relates the whole images.

Bao and Xu's Method

Bao and Xu [Bao99] use an approach based on wavelets for registering panoramic images. In their method, the images are decomposed in the complex wavelet domain and an edge-preserving visual perception threshold is applied to extract features. These features are matched by the similarity of wavelet coefficients, and finally these correspondences are used in a Levenberg-Marquardt non-linear least-squares algorithm to calculate the transformation. To reduce the high computational cost of this method, they use a hierarchical method to repeat the process at varying resolutions.

The problems with their method are the following.

- Their approach assumes no tilting and rolling of the camera in order to make use of the property of scaling and translation invariance of the complex wavelet transform. Therefore their approach is restricted to pure panning.

- Although thresholding eliminates many points, the remaining edge points still comprise a large number of features to match. For each edge point of one image, they restrict the search for a match in the other image to an $N \times N$ neighbourhood about that point. If N is chosen too small, the method may fail to find the correct match. If N is too large, the method is prohibitively computationally expensive. Thus, in practice, their methods assumes a good initial alignment is given.

- False matches are unavoidable when seeking correspondences, however robust a matching metric is claimed to be. False matches are particular harmful

when calculating the image correspondence transform. In their method, no measure is taken to exclude false matches.

- They use a non-linear least-squares algorithm to find the parameters of an 8-parameter perspective transformation between each pair of images. This requires a reasonable estimate of the transformation as a starting point. When the displacement between the initial images is large, this also causes them to need to manually provide an initial alignment.

- Their approach performs no gap closing or other refinement of the final panorama map.

Xiong and Turkowski's Approach

Xiong and Turkowski's work [Xiong98] is another application of computer vision techniques to building a full-view panorama from images captured from a fixed viewing position. Their approach has a pairwise registration method as well as a camera calibration and global optimization scheme. A gradient method is used for image registration, in which they introduce two extra exposure parameters to allow for exposure differences between adjacent images. The initial transformation parameters are estimated by a correlation based linear search. These parameters are then optimized by simulated annealing. They also use simulated annealing in their camera calibration algorithm where the presence of many redundant overlapping images is required. One problem of this non-linear optimization procedure is that it can easily become stuck in a local minimum. The initial camera calibration parameters are interactively provided. In regions where several images overlap, they perform a Laplacian-pyramid-based blending to determine the pixels of the final panorama, where they use a separate weighted average on each pyra-

mid level with different overlap lengths. Their approach requires a large amount of overlap, resulting in multiple overlaps of many images in the same region, which increases the expense of panorama construction. The user must also take more source photographs.

Peleg and Rousso's work

The above approaches to building panoramas assume that all images are taken from a single viewpoint, i.e. the camera is only rotated about its optical center. Such images fall in the category of ordinary perspective imaging, and can be combined to create the entire viewing sphere or cylinder centered at the common viewpoint. Further related, but less relevant work is also given in [McMillan95, Szeliski96, Irani95]. Another different approach is to use images taken with a camera moving along a smooth path, i.e., multiperspective imaging. This constructs a panorama which can be reprojected from its manifold representation [Peleg97, Rousso97, Peleg00].

Their methods create a panorama with an image strip [Peleg97, Rousso97, Peleg00]. The input images are captured from a smoothly moving camera, which may be pointing in the direction of motion in the special case, or in a direction orthogonal to it. In the latter case, the camera may also pan in the normal plane to the direction of motion. Assuming a known camera motion and a calibrated camera, thin strips to avoid perspective effects are taken from the input images and are placed, after warping, onto the mosaic manifold. The manifold is chosen in such a way that the optical flow becomes approximately uniform. The images are projected onto a pipe surface whose spine is the camera path. Any derived new image is then found by re-project from the pipe surface onto a suitable plane. The method is an approximate approach that requires a high frame-rate and small

depth differences in the real scene. The resulting panoramas can be interpreted as texture maps on a 2D manifold embedded in 3D. Because of the complicated 3D geometry of the manifold, rendering realistic planar images from the panoramas requires involved computations, and produces results of mediocre quality.

Other work on multiperspective panoramas can be found in [Rademacher98, Wood97].

Errors in Panorama Construction

In Kang and Weiss's work [Kang99], the authors have given a study of errors in building a panorama caused by errors in the camera's intrinsic parameters, e.g. focal length and radial distortion coefficient. Their conclusions are that the effect of focal length error is more significant than radial distortion and that errors in assumed focal length produce smaller relative errors in length of the composite cylindrical panorama. Note that the amount of overlap of adjacent images is determined in part by the focal length. Thus, they suggest that it is possible to correct the focal length by iteratively computing a new focal length from the composited panorama length. This method was also used by Shum [Shum00], as previously noted. This method of correcting focal length causes aspect ratio distortion.

Commercial Products

Besides a growing number of research papers, some commercial and free products exist for panorama construction and viewing, with some degree of recognition and success. Notable examples of such commercial products are LivePicture(MGI)'s Photovista [Photov], Apple's Quicktime VR [Qickvr](already discussed), and In-

finite Pictures' SmoothMove [Smoothmove]. Others are Terran Interactive Inc.'s Electrifier Pro [Terran], Enroute Imaging's Quickstich [Enroute], IBM's PanoramIX [Hotmie], Interactive Pictures' IPIX Multimedia Builder [IPIX], Panavue's Visual Sticher [Panavue], PictureWorks' Spin Panorama [pictureworks], and Web3D Consortium's VRML-browsers [VRML].

Some of these commercial products provide automatic stitching tools, such as Quicktime VR, Photovista and PanoramIX. We have tried them on various images, and overall experience is that their automatic registration capabilities are not very reliable.

Other related work in rending arbitrary view from image-based virtual reality model can be found in [Chang97, Chen97, Coorg98, Debevec96, Gracias00, Kanade97].

Summary

In summary existing methods suffer from one or more of the following shortcomings:

- Need for special hardware for taking photographs.
- A camera with known focal length.
- Perspective effects are ignored, and only an affine transform is determined between each pair of images.
- Tilting and/or rolling of the camera is ignored.
- An assumption is made that relative depth differences in the scene are negligible.

- Excessively large overlaps are required between adjacent images.
- Initial registration is done interactively.
- If registration is done automatically, no method is used to prevent false feature matches from causing incorrect registration.
- A global search is used for initial registration, or too many features are used causing registration to be slow.
- Fine registration can fail if initial registration, or assumptions about e.g. hardware, are too inaccurate.
- Non-linear parameter optimization procedure is sensitive to local minima and computational expensive.
- Panorama construction is not treated as a global problem, and mismatches between final and initial images remain, or are not properly corrected.
- The generation of new perspective views from the panorama representation is somewhat inaccurate.

We aim to produce a method for constructing panoramas under the assumptions given earlier, which do not have these deficiencies.

1.4 Thesis Contribution

In meeting the goal outlined at the end last section, this thesis reports a new approach to panorama construction, and the following novel contributions are made.

- An improved algorithm for high curvature point detection used in feature matching.

- A gradient and a shape based metric for matching features.
- A two-step registration process using features for initial registration, and a gradient method for further fine registration if needed.
- An iterative approach with linear steps to perspective transformation parameter optimization for pairs of adjacent images.
- A 5-parameter model for gradient based fine registration.
- An analysis of smooth factor in fine registration for enlargement of registration scope.
- Cylindrical warping methods which allow tilting and rolling to be present.
- Theoretical analysis of the effect of focal length error on the panorama end mismatch.
- An new approach to gap closing by iteratively adjusting the focal length and *panning* angles.
- Corrections to the final panorama which allow for tilting and rolling.

1.5 Thesis Outline

The rest of the thesis is organized as follow: Chapter 2 is an outline of this research. In Chapter 3, the feature-based registration approach is first described; while in Chapter 4, the gradient-based fine registration methods are discussed. Chapter 5 presents methods for constructing and tidying of final panorama. Examples obtained by the method in this thesis are demonstrated in Chapter 6. The last Chapter summarizes this research and discusses future work.

Chapter 2

Solution Outline

In our work, we study the building of a single nodal panorama, that is, the panorama constructed from a single viewpoint, also called the "center of projection". A panorama can be viewed as a projection of the scene onto a cylinder or a sphere through this viewpoint. We suppose we have obtained a series of overlapping natural photographs taken with a handheld camera, from a fixed viewpoint and with the same focal length. The panoramic image is created by determining the relative transformations between adjacent images in the sequence. Each image of the sequence is then projected onto a cylindrical surface whose radius is an initially estimated focal length (see Figure 2.1). Blending between overlapping source images is performed to construct the cylindrical image.

Since we assume that two adjacent images are taken from approximately the same location, corresponding image points in the two photographs are approximately in perspective correspondence. Hence we first find such a correspondence between a number of feature points from the two images and then use an optimization technique to find the perspective transformation that best relates the two images. A gradient based fine registration is further performed if necessary.

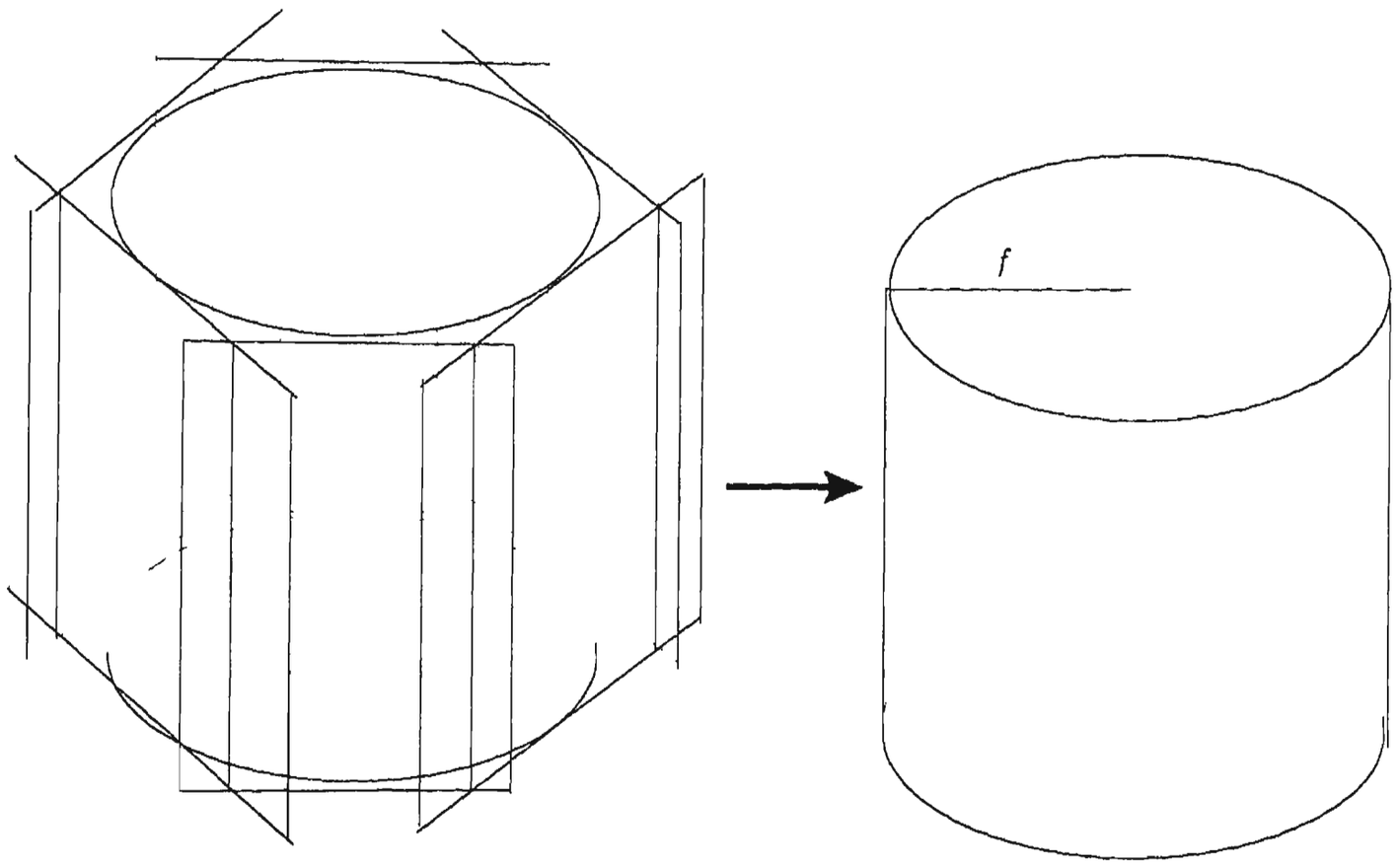


Figure 2.1: Cylinder model

Briefly, our approach comprises seven main steps:

- Selecting feature points in each image.
- Automatic identification of corresponding feature points in adjacent images.
- Computing a perspective transformation relating a pair of adjacent images from the matched features.
- Using a gradient-based fine registration procedure to improve registration, where if the number of feature points is insufficient or the registration error is too big.
- Initial panorama building by finding the mappings of the images onto a cylinder, to determine the panorama size and shape.
- Corrections to the transformations for incorrect focal length, camera tilting and rolling, and deskewing, in order to close the end mismatch.
- Blending the source images to produce the final panorama.

The first three steps are performed for each pair of adjacent images, and the fourth step is carried out only when further improvement of registration is needed. The fifth step maps the images onto a cylinder. Because the transformations are only found pairwise in these steps, a tidying process is used to make the transformations globally consistent. Finally, the panorama is constructed by blending the source images. A more detailed outline of each step is given next.

2.1 Image Registration

To stitch a sequence of images, we need to find the transformations between each pair of adjacent images. The first four steps of our approach solve this pairwise registration problem. We begin with our feature-based registration algorithm to find a set of correspondences and compute a transformation to represent the transformation. If necessary, we then invoke a gradient-based fine registration algorithm to improve the result.

2.1.1 Feature Based Registration

In our feature based approach, features are first identified in each of the two images, and corresponding features are matched according to some similarity metric. An optimization procedure is then used to compute a transformation that aligns features from one image with corresponding features in the other image.

The feature points we use are significant high curvature points, or referred as corners for short. To detect corners, we first convert the color image to a gray scale image and use a Canny edge detector to extract edge images from these gray images. Secondly, a threshold is used to convert the edge image to a black and

white image. Thirdly, a thinning algorithm is performed on this image. Finally, an *improved* high curvature point detection algorithm is applied. In the process, parameters are chosen so that only significant corner points are retained. The significant feature points are considered to be all corners in the minimum overlap region of the right image and then matches are sought from the whole left image.

The matching of correspondences is performed using a preliminary gradient threshold to reject impossible matches and two novel template matching metrics, a gradient metric and a shape metric, which are designed to tolerate both intensity differences and perspective distortions. Finally, false matches are discarded by a clustering procedure.

Once a set of correspondences has been obtained, initial estimates of camera rotation pan and tilt angles relating each image pair can be calculated. So an approximate alignment matrix from which the camera focal length can also be estimated. An initial transform matrix for each image pair that is used as a basis for refinement by optimization is then obtained from the transformation matrix generated from the pan and tilt angles and focal length.

An optimization procedure is then used to refine the matrix parameters. As a perspective transformation is involved here, the optimization is a non-linear problem theoretically. We have reduced the problem to a series of linear steps in our approach.

2.1.2 Fine Registration

In many cases, the above registration method gives good results for panorama building. However, occasionally, there may be insufficient features found to give good registration. At other times, a large residual will remain after least-squares

transformation estimation, which indicates that features have probably been mismatched. In either case, we use a further step of *fine registration* to improve the registration.

Our fine registration method is based on an optical flow motion detection technique [Horn81]. Because this method uses information at each pixel in the overlap region, it is potentially better than just using few feature points, but it can only recover a transformation differing by a few pixels in an image. The initial estimate of the camera transformation provided above, while not being good enough for direct use, *is* a good enough starting point for refinement by an optical flow method.

Two particular types of registration model are typically used in optical flow registration methods. In the first case, a general 8 parameter projective model is used to relate the motion between a pair of images [Bergen92, Szeliski97]; in the second case, a more restrictive model assumes a camera of known focal length is used and the camera only rotates between frames giving a 3 parameter model [Bergen92, Szeliski97].

In practice, we have found neither of these approaches is entirely satisfactory. The 8 parameter model is time consuming to compute, and also can produce large distortions in the registered images far from the overlap region, as it allows too general a type of transformation. While the 3 parameter model is simpler, it has the disadvantage of not allowing for errors in the focal length, which causes blurring in the final composited image. Hence, we use a novel 5 parameter model that is based on the rotation-only 3 parameter model but also allows recovery of the focal length.

Finally, we note one further refinement. Optical flow methods are based on the assumption that image intensities vary smoothly across the image, which is

clearly not true near edges. We thus smooth each image using a Gaussian filter before applying the optical flow model. This has been found to further improve the results.

2.2 Initial Panorama Building

Having initially registered individual images pairwise, we now need to stitch them together to form a complete cylindrical panorama. First, transformations are found which allow the aligned images to be projected onto planes tangential to the viewing cylinder, and warping matrices are computed for mapping the planes onto the cylindrical surface. This step provides the size and shape of the composite panorama, information which is needed by the following tidying process.

To smooth out intensity differences between overlapping source images, we interpolate the intensity of each pixel linearly from the two contributing images according to the pixel's distance from the borders of the overlapping region. (As we assume that there is small overlap between adjacent images, we neglect here the possibility of three or more images overlapping, although our practical implementation does also successfully handle such case.)

2.3 Panorama Tidying

Because the steps above register images pairwise, i.e., locally, a gap or overlap—an end mismatch—may remain between the last and first images. The most satisfactory way of dealing with this would be to treat panorama construction as a global problem using all images simultaneously. However, formulating the problem in this way is difficult, and even if done, is likely to be computationally very expensive,

and quite possibly hard to solve robustly. Thus instead, we take the approach of making corrections to the initial pairwise transformations to resolve the end mismatch.

Four separate processes are used in correcting end mismatch. Previous steps assume the tilt and roll of the initial image are zero. As a result, if these angles are non-zero, the panorama produced will form a strip on the cylinder in the form of a sine curve or a helix respectively. This is detected, and the initial tilt and roll are corrected accordingly.

If the initial focal length estimate is incorrect, the whole panorama will not exactly wrap around the cylinder, but there will be a gap or overlap. This can also be detected and used to correct the focal length.

Finally, even after these corrections, the overall unwrapped panorama may not quite form a rectangle because of remaining minor errors in the pairwise transformations. A general perspective transformation is used to map the four corners of the panorama exactly to the four corner of the appropriate rectangle, ensuring a good match between the last and first images. Essentially, remaining errors are thus distributed over the whole panorama rather than being concentrated in one place, reducing their visible effects.

2.4 Panorama Composition and Blending

Having updated the pairwise image transformations, the warping matrices for mapping the images to the cylinder are also updated. For each pixel of the destination panorama image, these transformations determine which pixel of which source image corresponds to it. Pixels in the overlap region are determined by a weighted blending function which interpolates the pixel intensity of source images according

to destination pixels' position in the overlap region.

Chapter 3

Feature Based Registration

To stitch together a sequence of photographs to build a panorama, we need to align or register each pair of adjacent images. Since we assume all photographs are taken from a fixed single location, the problem of image registration becomes one of finding a *perspective* transformation which aligns two adjacent images. As we noted in an earlier chapter, there are basically two categories of image registration approach: one is feature-based which we will address in this chapter; the other, which we will discuss in the next chapter, is the gradient or optical flow approach.

In our feature-based image registration approach, salient feature points are extracted from image pairs by our high-curvature point detection algorithm which is designed to precisely locate the position of salient points. Corresponding salient points are then matched by our robust matching metrics, a *gradient metric* and a *shape metric*. An iterative optimization procedure, using linear steps, then minimizes the Euclidean distance between corresponding points to find the perspective transformation that relates pixels in one image to pixels in the other.

3.1 Overview

Image registration is concerned with the establishment of correspondence between images of the same scene, and determining the geometric transform that aligns one image with another. A traditional method of automatic registration is an exhaustive search using correlation of two images in the spatial or frequency domain. However, the computational complexity of this approach makes it impractical even for medium size images, especially when a general transformation, such as a 3D camera rotation, is involved. To reduce the cost of search, registration is often computed using salient feature points instead of the whole image area. After salient points are found in each image, correspondences are sought to match salient points in a pair of images. The correspondences (tie-points) are then used to estimate the transformation parameters aligning the images via an optimization scheme. This is the general idea of the feature-based registration approach.

There are three parts to the feature based approach: extracting salient points in each image, finding correspondences between them, and estimating the transformation parameters from the correspondences.

The first two parts consist of extracting salient feature points in images and matching feature correspondences between images. The task can be done manually, but it is very tedious to specify a complete, or even a sufficient, set of correspondences. Therefore automatic methods are exploited. In an automatic approach, salient feature points are first detected in the regions of interest of each image in the pair. Typically used features are corners, line intersections, high curvature points and centroids; instead of points they may also be higher-level structural and syntactic elements. Using features removes extraneous information contained in images and reduces the amount of data to be evaluated. In extracting salient

feature points, we want the features to be unambiguous, precisely located and invariant under local distortion of the image.

Matching features is the task of finding a corresponding feature in one image for its counterpart in the second image, which is a challenging task. Contributing factors to this difficulty include the lack of image structure, repeated or similar elements in the image, object occlusion, and acquisition noise, which are frequent in real imaging applications. Various metrics have been developed to decide the similarity of features. Among them the most commonly used are normalized cross correlation (NCC) and sum of squared differences SSD. The NCC statistically compares the similarity of two image windows of the same size centered at the feature points [Pratt74, Gonzalez93], while the SSD sequentially computes the absolute squared difference between the two image windows. Other metrics include coincident-bit-counting [Chiang93] and ordinal measure [Bhat98]. The measured primitives can be image attributes other than raw pixel intensities, such as Fourier spectra, wavelet coefficients, contour-chain-code and moments, for example. These metrics, especially NCC and SSD, are optimal when the features of interest to be compared are identical. However, image preprocessing is required to improve the performance if a significant amount of noise is present. Furthermore, false matches may occur if there are significant differences between two images due to noise and distortions in the images. Thus, a global consistency check is needed to exclude mismatches arising from such causes. In the whole registration procedure, a robust matching procedure is a critical issue.

In the third task, transformation parameters are estimated through approximation from the matched feature correspondences, in which parameters of the transformation are found so the matched points are aligned as nearly as possible. In least squares approximation, the sum over all corresponding feature points of the

squared differences after alignment is minimized. The minimum can be determined by setting the partial derivatives to zero, giving a system of linear or non-linear equations. To find the best approximation, the number of matched points must be sufficiently greater than the number of parameters of the transformation so that sufficient statistical information is available to make the approximation reliable. Individual matches are likely to be somewhat inaccurate, but taken together they should contain sufficient information to determine the transformation. The transformation modelled can be an affine, a perspective or a polynomial one. In our application to building a panorama, the transformation to be found is a perspective one. Thus, the parameter approximation is in principle a non-linear problem. An efficient solution to this issue is required.

Based on the above general idea of feature-based approaches, we derive special-purpose methods for image registration to suit the panorama construction problem. These must be robust against perspective distortion and must tolerate brightness variations. In the best case, this approach should provide an accurate image registration, but in the worst case, it should at least provide an adequate coarse alignment to be used as the input for subsequent fine registration.

The rest of this chapter is organized as follow: In Section 3.2, we outline our feature-based registration algorithm. Section 3.3 describes feature extraction, while Section 3.4 addresses feature matching. The determination of transformation parameters is presented in Section 3.5. Experimental results are given in Section 3.6. The last section is a summary of the chapter.

3.2 Algorithm Outline

As our aim is to devise a fully automatic solution to panorama building, a feature-based approach is adopted in our work as the first step towards the final registration. Given two adjacent images with adequate overlap, we perform the following steps for finding features:

- *Edge detection:* Edge images are extracted from each image using a Canny edge detector.
- *Edge thinning:* Thinning of each edge image is performed.
- *Salient point detection:* Salient points are selected from each image by finding points of high curvature.

The next two steps find matching features:

- *Candidate match finding:* A local measurement is used to find candidate matches between salient points in the two images.
- *Global consistency checking:* A global method is used to remove inconsistent matches from the list of candidate matches.

The last two steps find the transformation parameters:

- *Initial parameter estimation:* We estimate initial focal length, initial panning and tilting angles, and compute an initial transformation matrix.
- *Accurate parameter determination:* We perform an optimization procedure to refine the transformation parameters.

We now discuss each step in detail.

3.3 Extracting Salient Feature Points

Features, or salient points, are specific pixels in an image which have easily distinguished meaningful characteristics in the scene. Features play an important role in the effectiveness of feature-based registration approaches since finding correspondences and thereafter computing the required transformation both depend on them. Features should be selected using points that (1) are highly distinguishable and unambiguously localizable in each image; (2) contain sufficiently discriminating information for matching; (3) are adequately tolerant of local distortions. Note that in our application of stitching photographs for building a panorama there are possibly big perspective distortions and brightness differences between the images. From these observations we can deduce that local maxima of curvature points on edge contours in the images are a good choice to be used as feature points. In the following three subsections, we describe our method of determining maximum curvature points. We use a Canny edge detector for edge extraction, an onion peeling algorithm for edge thinning, and an improved maximum curvature detector for feature point extraction. We now begin with a description of edge detection.

3.3.1 Edge Detection

First we convert the input color images from RGB format into grayscale images. According to the human visual perception of the color spectrum, the eye is most sensitive to the green component, and less so to red and blue colors. We may convert from R, G, and B color bands to give a single gray intensity for each pixel using

$$Y = 0.30R + 0.59G + 0.11B.$$

By converting color images into grayscale images, both the time and space costs of our method are reduced. All processes thereafter are based on this gray scale image, until the final panorama is assembled. Examples of this conversion are shown in Figures 3.1 and 3.2. Figure 3.1 contains color images, and Figure 3.2 contains the converted grayscale images.

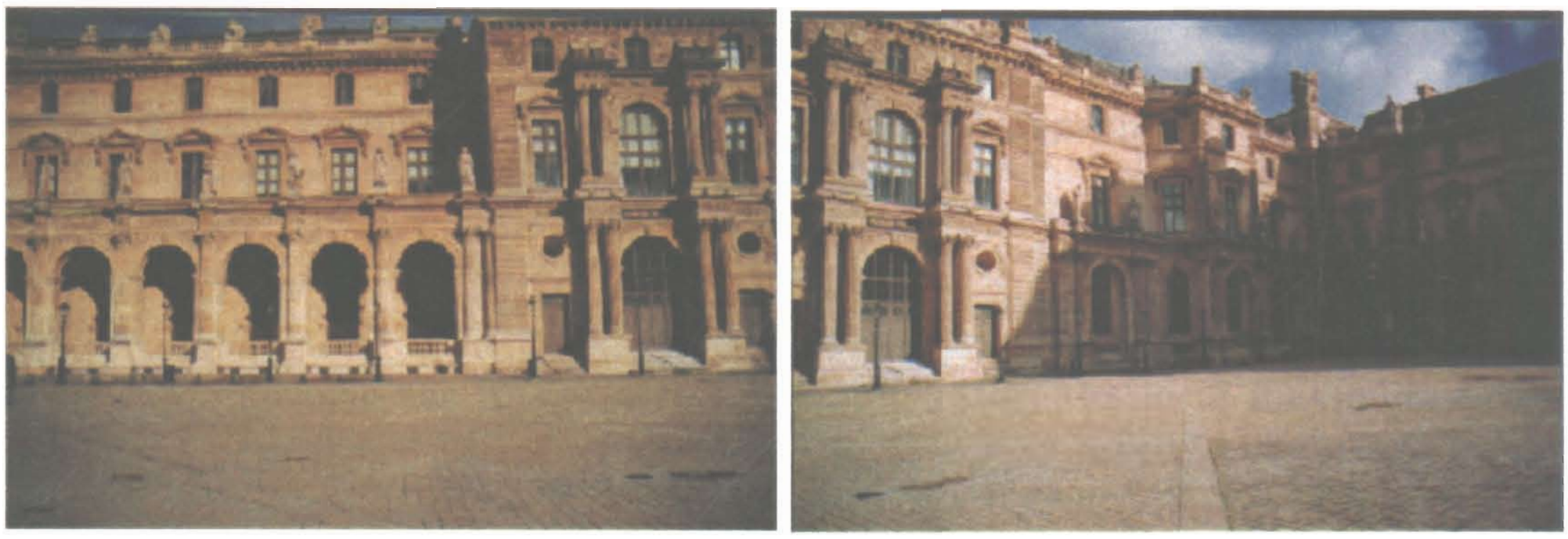


Figure 3.1: Images with brightness differences and perspective distortion.

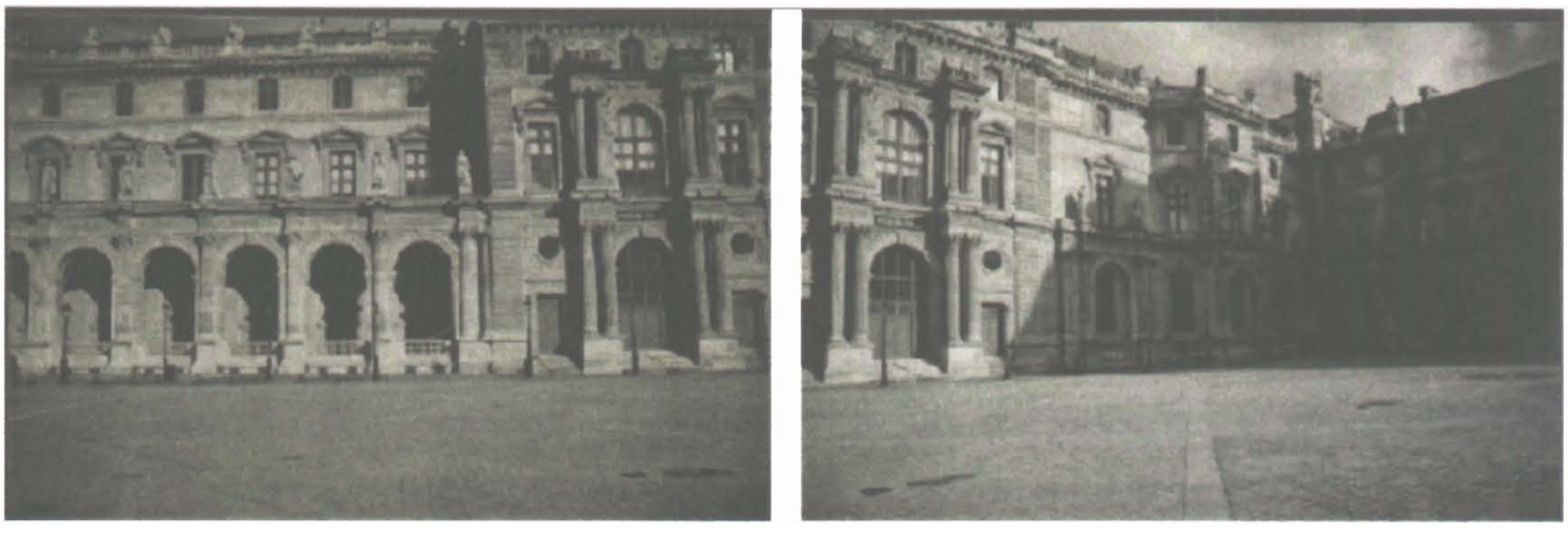


Figure 3.2: Converted grayscale images

Next, an edge image is formed from each image using a Canny edge detector [Canny86]. The Canny edge detector is a significant and widely used contribution to edge detection techniques. Briefly, the concept of the Canny edge detector is as follows.

Suppose G is a 2D Gaussian

$$G = exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) ,$$

and G_n is an operator which is the first order derivative of G in the direction $\mathbf{n}$

$$G_n = \frac{\partial G}{\partial \mathbf{n}} = \mathbf{n} \cdot \nabla G \tag{3.1}$$

where $\mathbf{n}$ is the normal perpendicular to an edge. Assuming g is the image, $\mathbf{n}$ can be estimated as

$$\mathbf{n} = \frac{\nabla(G * g)}{|\nabla(G * g)|} .$$

If we now convolve the image with the G_n operator, the local maximum in the direction of $\mathbf{n}$ is the edge location, which is given by

$$\frac{\partial}{\partial \mathbf{n}}(G_n * g) = 0 .$$

Combining with Equation (3.1) gives

$$\frac{\partial^2}{\partial \mathbf{n}^2}(G * g) = 0 . \tag{3.2}$$

This operation is often referred as *non-maximal suppression*: it shows how to find a local intensity maximum in the direction perpendicular to the edge.

Having thus determined the edge location, the strength of the edge (the magnitude of the gradient of the image intensity function g) is measured as

$$|G_n * g| = |\nabla(G * g)| .$$

A version of the algorithm is implemented in our system, where we leave the deviation σ to be adjustable by the user. In performing the edge detection, we must use a sufficiently large deviation to smooth the images to avoid spurious edges, so that only significant edges are retained for further processing, while undesirable

effects of noise and texture are minimized to remove extraneous information and reduce the amount of data to be evaluated. An example of Canny edge detection is shown in Figure 3.3, where σ is 2.5. The value of σ affects the number of features detected afterwards. A lower value keeps more edges and thus extracts more salient points, while a higher value helps to smooth out distortions and noise. By our experience, the value ranging from $2 \sim 5$ is adequate for various images.

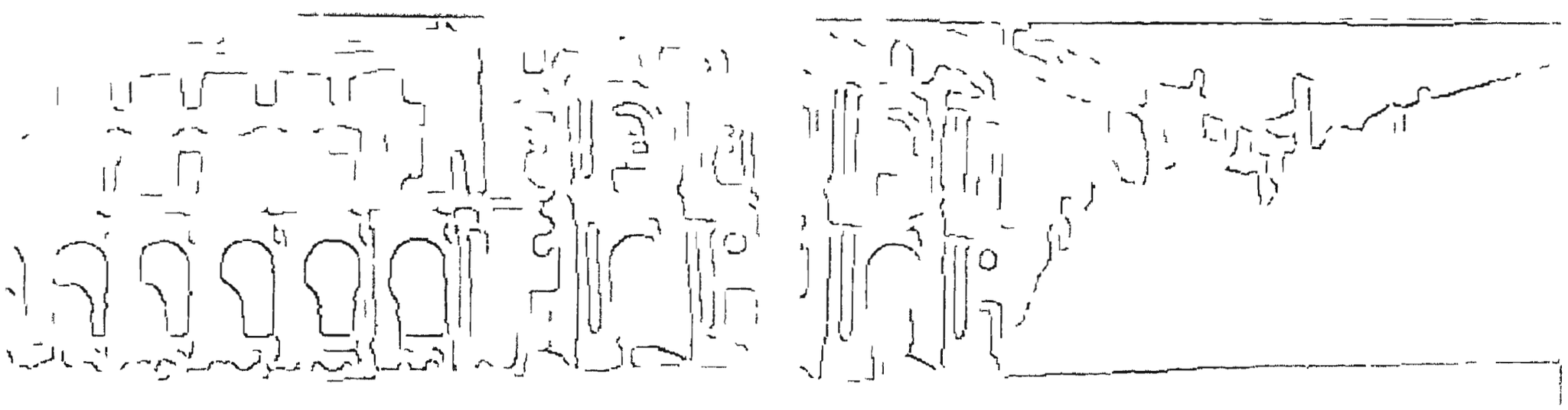


Figure 3.3: Grayscale edge images

We then use a threshold T_g to reduce the grayscale edge image to a binary image. Like deviation, a lower value of T_g retains more edges which in turn leads to more salient points. By our experience, a value of T_g ranging $0.01 \sim 0.3$ is sufficient to include enough significant edges. The value measured is what after the maximum edge intensity in the image is normalized to 1. Normally we set this threshold to 0.2. An example of this threshold operation is shown in Figure 3.4.

3.3.2 Edge Thinning

After thresholding the edge images, we perform edge thinning. This step is required by the later maximum curvature detection which is based on tracing edges of one pixel width. The edge thinning is done with the aid of a set of masks [Guo92,

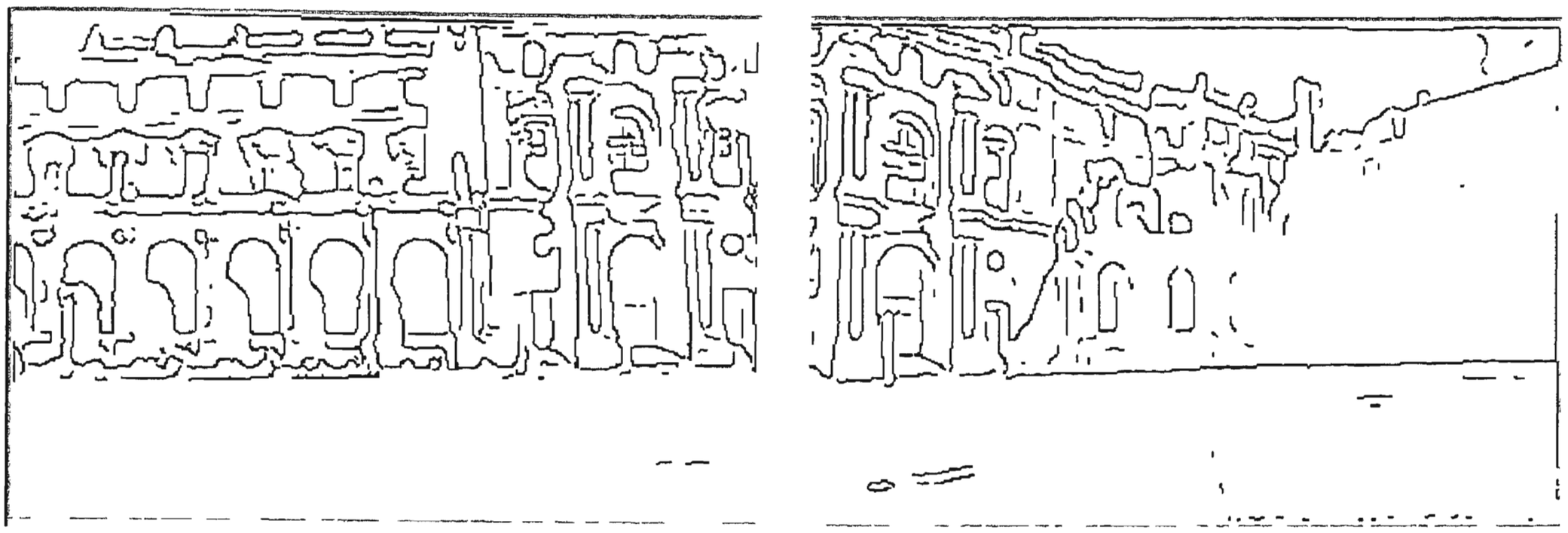


Figure 3.4: Binary edge images

Satoshi87]. Each mask uses the values of the pixels adjacent to the pixel at the center of the mask to determine whether to keep or discard the pixel as an edge point. The masks are applied recursively in an "onion peeling" manner until there is no further change. We omit the detail of how to set these masks, see [Guo92] for reference. An example of this thinning step is shown in Figure 3.5, the image before thinning is shown in Figure 3.4.

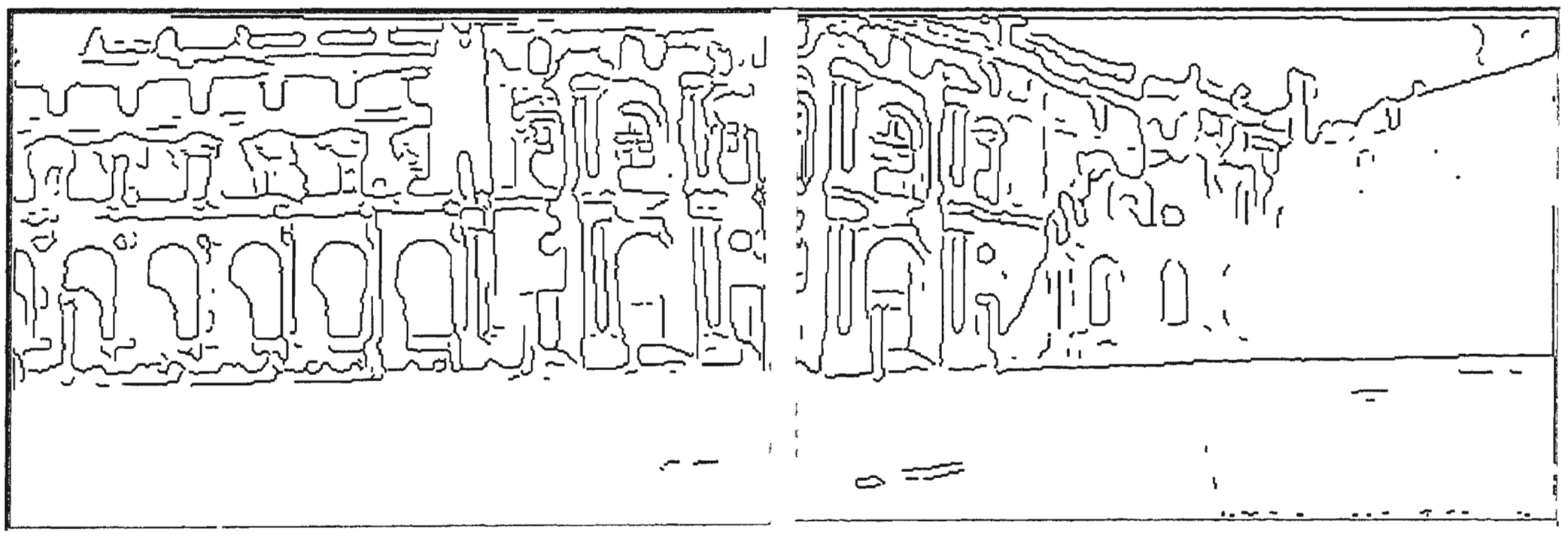


Figure 3.5: Binary edge images after thinning

3.3.3 Feature Point Detection

We now wish to find the feature points in each image. A search is made along each edge to find points of high curvature as salient feature points. Our high-curvature detector is adopted from Li [Li95], but is improved to locate the corner points more accurately.

In [Li95], a high-curvature point is detected using the chain code representation for the curve. The measure of curvature at the *i-th* point is defined to be, modulo 8,

$$c_i = \max_{1 \le j \le 3\sigma} \{\max\{|a_{i-j} - a_{i+j}|, |a_{i-j} - a_{i+j-1}|\}\}$$

where a_i is the chain code from point $i-1$ to point i, and σ is the standard deviation of the Laplacian of the Gaussian used in the edge detector. In his method the *i-th* point is chosen as a feature point if (i) c_i exceeds a curvature threshold C; and (ii) if c_i is a maximum value in a neighborhood of length $2l$ along the line segment, i.e.,

$$c_i \ge c_k \text{ for all } k \in [i-l, i+l],$$

where l is a length parameter chosen to be $l = 4\sigma$. However, this approach does not work well in locating feature points when several points in $[i-l, i+l]$ have the same maximum curvature. This disadvantage becomes worse if larger values of l are used.

In light of the above problem, we have made improvements to this method to make feature point localization more accurate. Firstly, we redefine c_i to be

$$c_i = \max_{1 \le j \le l} \{\max\{\delta_{i,j}, \Delta_{i,j}\}\},$$

where

$$\delta_{i,j} = \angle(\mathbf{p}_{i-j+1} - \mathbf{p}_{i-j}, \mathbf{p}_{i+j+1} - \mathbf{p}_{i+j})/2l,$$

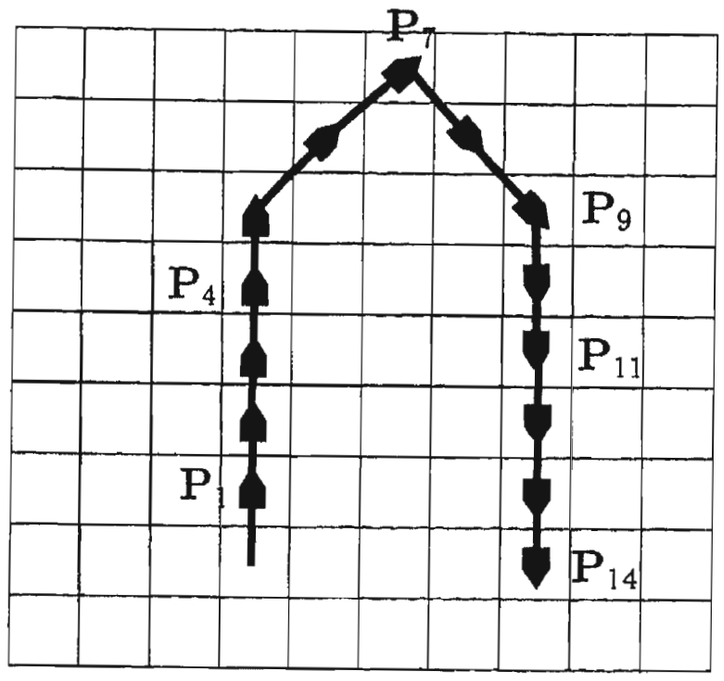


Figure 3.6: Finding point of highest curvature

$$\Delta_{i,j} = \angle(\mathbf{p}_{i-j+1} - \mathbf{p}_{i-j}, \mathbf{p}_{i+j} - \mathbf{p}_{i+j-1})/2l,$$

and $\mathbf{p}_i$ is the i^{th} point.

Secondly, we introduce another quantity d_i at the i^{th} point to record the least j which gives the maximum in the expression for c_i:

$$d_i = j,$$

where j is the least index such that

$$c_i = \delta_{i,j} \text{ or } \Delta_{i,j}.$$

When c_i attains its maximum value in $[i-l, i+l]$, the *i-th* point is taken as the feature point. When there are more than one such maximum point in $[i-l, i+l]$, we take the *i-th* maximum point as the feature point if it also has a minimum value of d_i among all maximum points.

We illustrate the idea in Figure 3.6. Let $l = 5$. With the definitions above, points $\mathbf{p}_4$ to $\mathbf{p}_9$ have the same curvature $c_i = \frac{\pi}{10}$ and all are points of maximum curvature along the line segment from $\mathbf{p}_0$ to $\mathbf{p}_{14}$. But the exact position of the feature point is at $\mathbf{p}_7$, which has the least value of d_i.

Figure 3.7: Salient feature point extraction

We have also modified Li's algorithm so as to be able to select and include T-junctions as feature points.

A critical issue in further processing is the number of feature points detected: for better accuracy, a correspondence based matching method needs more features; on the other hand, too many points will make the matching more time consuming. The number of feature points detected can be controlled by suitable choices of the threshold values C and l. We set $C = \frac{\pi}{4l}$. A lower value of l can extract more salient points. In practice we found a value of l ranging $5 \sim 20$ is adequate to extract enough features. By choosing a relatively small number of important features, matching can be done rapidly. Another issue to which attention should be paid is the interval between features. If the features are too closely located, they are apt to cause false matches later. We have set a threshold in our algorithm to only pick up features which are reasonably far apart. Figure 3.7 illustrates the results of our maximum curvature salient point detector, the salient points being marked by '+'. The parameter l is chosen to be 10.

3.4 Matching Feature Correspondence

In this section, we describe our method of finding matching features in a pair of images. First we introduce some concepts.

3.4.1 Basic Concepts

First we define correspondence. Let I and I' be two images of the same scene. An image point p in I and an image point p' in I' are corresponding points if they are projected from the same 3D point. A correspondence between two views is a mapping from one view to another such that each pair in the mapping is a pair of corresponding points.

Determining correspondence is performed using similarity metrics which are criteria to determine what types of feature matches are optimal. A good similarity metric should be robust with respect to various sources of noise in the environment. The choice of similarity metrics is one of the most crucial elements of the feature-based registration approach.

When searching for a match between a source window centered at a feature point and a target window, frequently used similarity metrics are based on the sum of squared differences (SSD) and normalized cross correlation (NCC). Let I_1 and I_2 represent the intensities of each window of size $n \times n$ pixels.

The sum of squared difference is given by

$$SSD = \sum_{i=1}^{n} (I_1(x_i, y_i) - I_2(x_i, y_i))^2.$$

This quantity measures the squared Euclidean distance between I_1 and I_2 and a value close to zero indicates a strong match.

The normalized cross-correlation is given by

$$NCC = \frac{\sum_{i=1}^{n}(I_1(x_i, y_i) - u_1)(I_2(x_i, y_i) - u_2)}{\sqrt{\sum_{i=1}^{n}(I_1(x_i, y_i) - u_1)^2 \sum_{i=1}^{n}(I_2(x_i, y_i) - u_2)^2}}$$

where u_1 and u_2 are the intensity means of each window. This quantity lies in $[-1, 1]$, and a value of one indicates a perfect match. Notice the cross-correlation is normalized so that local image intensity does not influence the measure.

Generally, the NCC metric is preferred to the SSD as it is invariant to a linear gray-level shift between otherwise perfectly matching windows, but SSD is computationally more efficient. The basic assumption used by these two metrics is that these windows represent the same location in the scene and have identical intensity distributions. However, the assumption is violated due to a number of physical phenomena because of which intensity data in windows around corresponding points can be inconsistent. For instance, when perspective distortion and illumination differences are present, the peak of the metrics may not be in the expected position. To deal with the problem, we now derive new matching metrics, and explain our matching procedure.

3.4.2 Preliminary Matching

Having found feature points in each image, locally consistent matching features are found in adjacent images. The width of the overlapping region is assumed to be at least 1/6 of the width of the two adjacent images, to provide sufficient matches to be found to allow registration to be performed. Thus, salient feature points are found in this 1/6 of the first image, and then their counterparts in the second image are sought over the entire second image, as the actual amount of overlap may be greater than 1/6. In searching for the counterpart of a feature point, we carry out a three step matching procedure which uses a preliminary derivative threshold to

rapidly discard poor matches, followed by a gradient template measurement and a shape template measurement for reliable detection of locally correct matches.

First order image derivatives have opposite signs on edges of increasing intensity and edges of decreasing intensity. We use this property to find candidate matches. Thus, the signs of first order image derivatives at each feature point from the first image and at each feature point from the second image are compared. This comparison effectively eliminates a large number of impossible matches by imposing that two potentially matching feature points have compatible gradient directions; only when they do are the pair entered into a list of candidate matches. This list is further assessed by two more sophisticated metrics, a *gradient template metric* and a *shape template metric*. In the gradient template metric, we measure normalized cross-correlation separately on the x and y partial derivative images. This measurement is much more robust than computing cross-correlations on raw intensity images

since derivatives represent the image structure and thus is in general more invariant to radiometric influences than does the raw intensity. This metric is also more robust than computing cross-correlations on edge images which have lost all directional derivative information. As images may have large perspective distortions in our problem, we also use a similarity metric based on edge shape to make feature point comparison more stable with respect to distortions. Details of these two metrics are presented below.

3.4.3 Gradient template metric

Let I and I' be two image windows centred at feature points P and P' in the two images, respectively. Other quantities for the second window will also be denoted

by $'$ as appropriate. Let $\nabla_x I$ and $\nabla_y I$ be the partial derivatives of intensity for window I. Let u_x and u_y be the means of the partial derivatives $\nabla_x I$ and $\nabla_y I$ over the window. We define the normalized cross-correlation *gradient template metric* between I and I' to be

$$NCC(P,P') = \frac{1}{2}\left(\frac{\sum_{i=1}^{n}(\nabla_x I(x_i,y_i)-u_x)(\nabla_x I'(x_i,y_i)-u'_x)}{\sqrt{\sum_{i=1}^{n}(\nabla_x I(x_i,y_i)-u_x)^2}\sqrt{\sum_{i=1}^{n}(\nabla_x I'(x_i,y_i)-u'_x)^2}}\right.$$
$$\left. + \frac{\sum_{i=1}^{n}(\nabla_y I(x_i,y_i)-u_y)(\nabla_y I'(x_i,y_i)-u'_y)}{\sqrt{\sum_{i=1}^{n}(\nabla_y I(x_i,y_i)-u_y)^2}\sqrt{\sum_{i=1}^{n}(\nabla_y I'(x_i,y_i)-u'_y)^2}}\right).$$

This $NCC(P,P')$ metric is first applied to all possible feature point pairs (P,P') in the candidate match list. Clearly, $0 \leq NCC(P,P') \leq 1$. Two threshold values T_1 and T_2 with $0 < T_2 < T_1 < 1$ are pre-specified. A pair for which $NCC(P,P')$ exceeds a threshold T_1 is provisionally accepted as a pair of tie-points, i.e. as corresponding feature points. Those pairs for which $T_2 < NCC(P,P') < T_1$ will also be provisionally accepted if they are shown to be a good match by the *shape template metric* described below. Any pairs for which $NCC(P,P') < T_2$ are rejected. In this $NCC(P,P')$ measurement, a value close to 1 indicates a good match. Practical experiments have shown that a value exceeding 0.6 is good enough to provisionally accept a match, while a value smaller than 0.3 is bad enough to warrant rejection. So, we take $T_1 = 0.6$ and $T_2 = 0.3$.

The exact values of these parameters are not critical, as the candidate matches will be further evaluated by another metric and mismatches will be excluded by a consistency check afterwards.

3.4.4 Shape template metric

Consider a small window W centered at a salient point P. The distance from each window boundary point to the nearest point of the edges in the window is

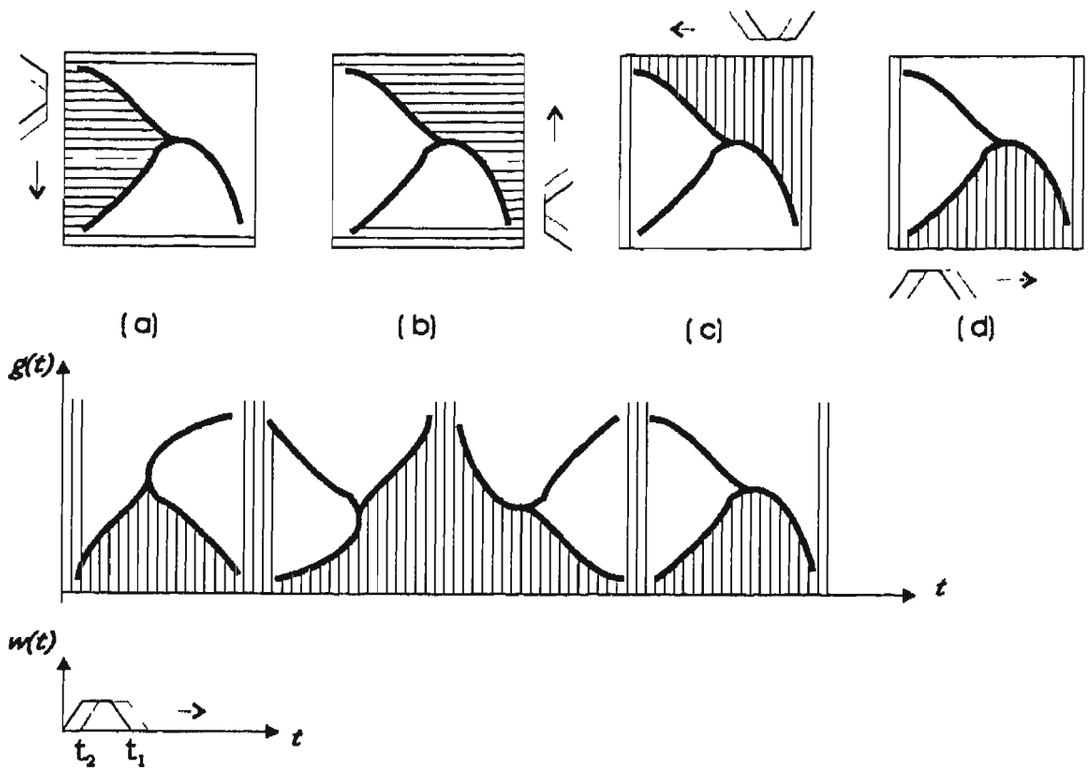


Figure 3.8: Shape feature vector

measured, and all such distances are used to characterize the image structure near P. As illustrated in Figure 3.8, these distances (hatched) from the four sides are concatenated, smoothed, and sampled to form a feature vector $f(x)$, where

$$f(x) = \sum_{t=0}^{t_1} w(t)g(xt_2 - t), \quad x = 0, 1, 2, ...m.$$

Here $g(x)$ is the distance from a point on the window border to the contour, $w(t)$ is a smoothing filter, t_1 is the filter length, and t_2 is the sampling interval. Then, given two template windows around P_1 and P_2 to be matched, the *shape template metric* D_{12} between them is the absolute difference between their feature vectors, i.e.,

$$D_{12} = \sum_{x=0}^{m} |f_1(x) - f_2(x)|.$$

A given P_1 is matched to P_2 if D_{12} is the minimum among all possible choices for P_2, and simultaneously, for P_2, D_{12} is the minimum for all possible choices of P_1.

3.4.5 Global match consistency checking

By this stage, provisionally matched pairs of salient points as determined by the gradient template metric and the shape template metric have been found. However, these metrics depend only on local image information in a neighborhood of a salient point, so a match that is acceptable only according to these metrics is not necessarily a true match when considered globally. To address this problem, we use a clustering technique to check which candidate matches yield mutually consistent information about the perspective transformation to be found.

Each pair of tie-points determines an approximate pan and tilt angle that relate the adjacent images. These two angles may be represented as a parameter point in a 2D parameter space. After representing all pairs of tie-points in the parameter space, only tie-points whose parameter points are in the largest cluster are accepted as true matches. Other tie points are regarded as unreliable and thus discarded. In performing clustering, a threshold d is set to compare the distance from a feature point to the center of a cluster. If the distance does not exceed this value, the feature is merged into this cluster. The center of a cluster is recomputed each time a new member joins in. The idea is demonstrated in Figure 3.9.

The clustering technique is a classical statistical method [Stockman82, Goshtasby85]. The method has a time complexity of $O(n^4)$, where n the is number of correspondences. This becomes prohibitive as the number of points grows. In our application, as will be discussed in the following section, it is possible to achieve a fine resolution of registration with a relatively small set of correspondences. As a result, the time taken for clustering can be kept low.

A final result of correspondence matching is shown in Figure 3.10, which shows the remaining matches left after the preceding steps, including global consistency

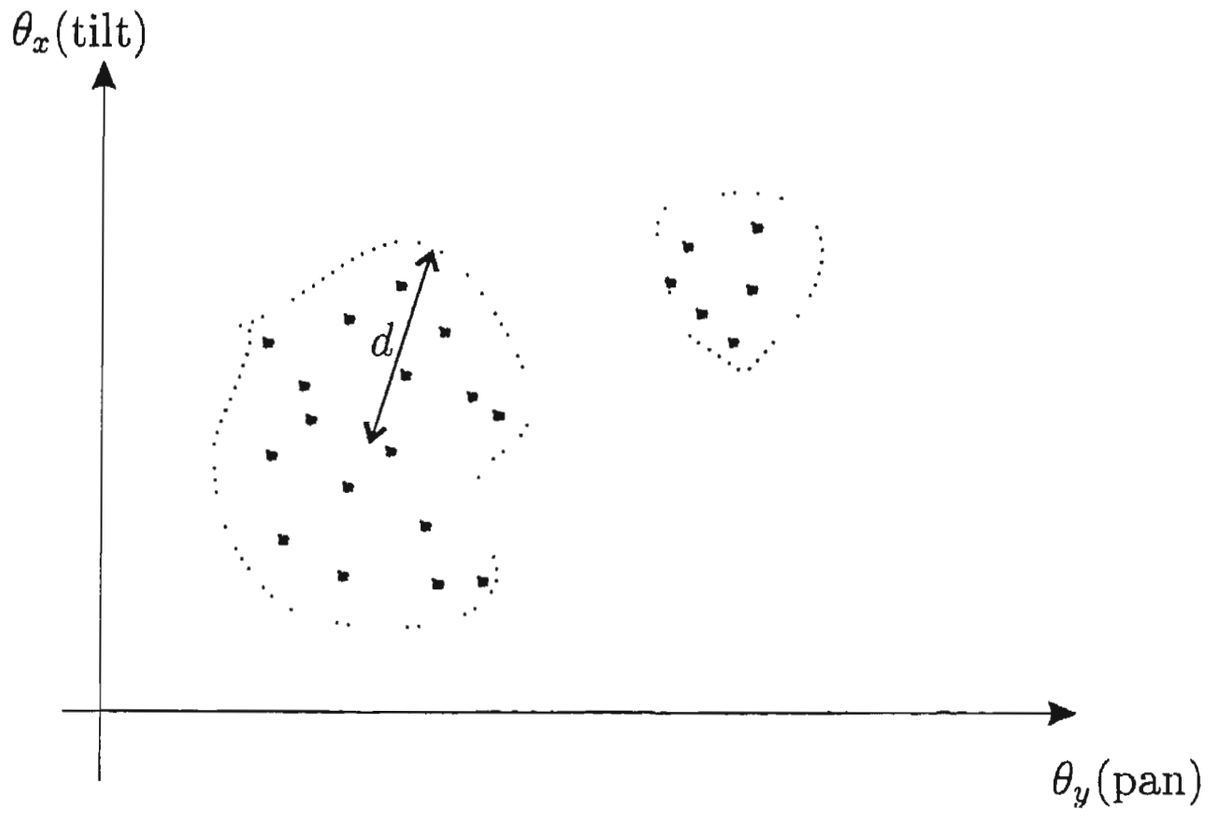


Figure 3.9: Clustering for global constancy check

checking.

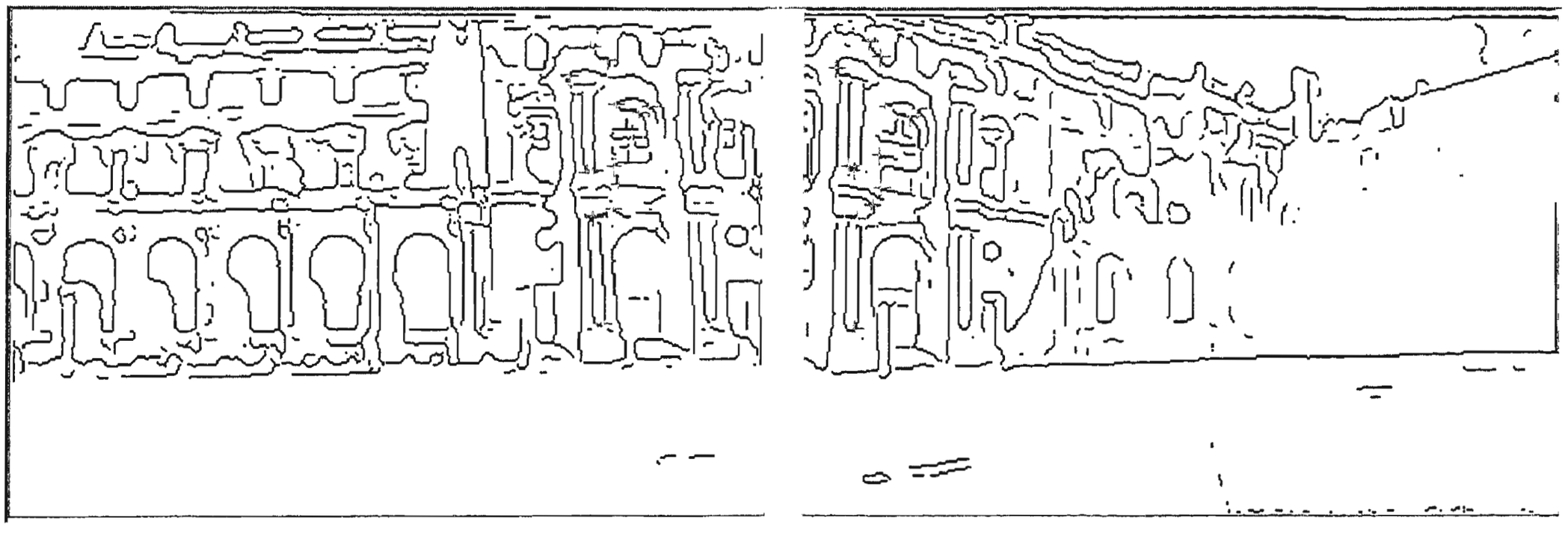


Figure 3.10: Tie-points identified on edges

After this global consistency check has been done, the perspective transformation between the two adjacent images is found using an error minimization procedure described in the next section.

3.5 Finding the Perspective Transformation

In this section we show how to find the perspective mapping M that aligns two adjacent images. The procedure consists of an automatic initial estimation of the parameters and an optimization procedure to improve the estimate.

3.5.1 The problem

Consider two initial images. Since the camera is assumed to be at a fixed location, the transformation M that relates two images is a perspective transformation, which can be computed from the correspondences between the salient points (x_i, y_i) in one image $I(x, y)$ and the matching salient points (x'_i, y'_i) in the other image $I'(x', y')$, $i = 0, 1, .., n$, where $n + 1$ is the number of feature matches.

The general model for a perspective transformation is an 8-parameter 2D homogeneous matrix M [Faugeras93, McMillan95]:

$$M = \begin{pmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{pmatrix}, \tag{3.3}$$

which maps points in one image into points in the other via

$$x'_i = \frac{m_0 x_i + m_1 y_i + m_2}{m_6 x_i + m_7 y_i + 1}$$
$$y'_i = \frac{m_3 x_i + m_4 y_i + m_5}{m_6 x_i + m_7 y_i + 1}.$$

The parameters can be found by minimizing the sum of squared Euclidean distance between the transformed points in one image and the unchanged points in the other image. Obviously, this is a non-linear optimization problem for the general model. The problem can be solved by using the Leveberg-Marquardt algorithm. However,

this approach is apt to become trapped in local minima and is computationally expensive. Furthermore, our special case of camera rotation does not allow a general perspective transformation. Thus, this general model contains extra free parameters which may lead to poor parameter estimates. We show next that the transformation in our application can be described by four unknowns.

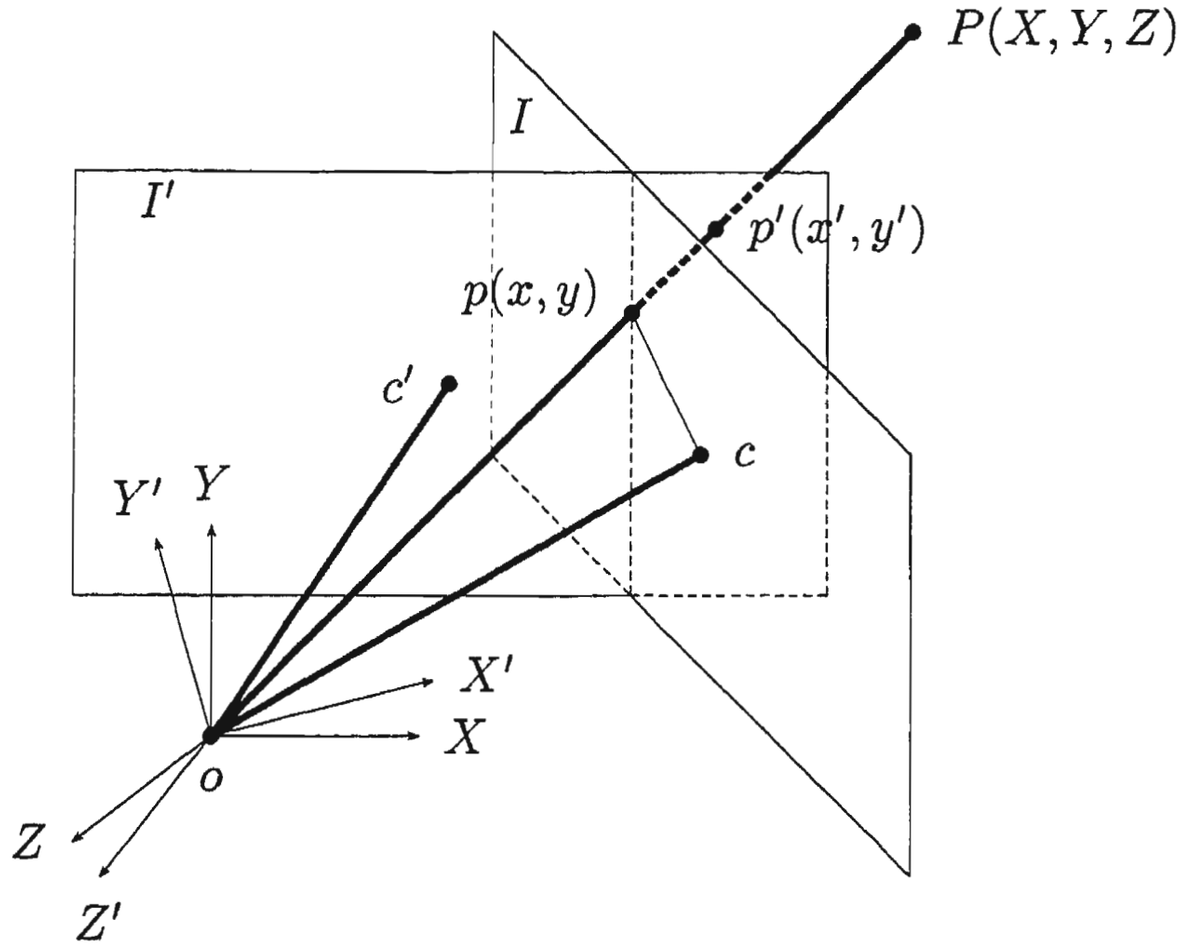


Figure 3.11: Perspective transform with images of camera rotation

Let θ_y, θ_x, and θ_z be the relative angles of camera panning, tilting and rolling, respectively, for the two adjacent images I and I'. Panning is (intentional) camera rotation about a vertical axis, tilting is (accidental) rotation of the camera towards the floor or sky about a horizontal axis, and rolling is (accidental) rotation of the camera about a horizontal line perpendicular to the image. Let f be the camera focal length, the distance between the centre of each image plane and the camera's optical center o; f is fixed but unknown. These four parameters θ_y, θ_x, θ_z, and f are to be found, and determine M.

Let (X, Y, Z) be a 3D world coordinate with its origin at o. Let (x, y) be a 2D image coordinate with its origin at the image center c, and for which the x and y

axes coincide with those of 3D coordinates. Suppose $P = (X, Y, Z)$ is a 3D object point. Its projection on the first image is $p = (x, y)$, and on the second image is $p' = (x', y')$: see Figure 3.11.

The camera rotation matrix is given by

$$R(\theta_y, \theta_x, \theta_z) = \begin{pmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{pmatrix} \cdot \begin{pmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (3.4)$$

which transforms the 3D point P to a new point $P' = (X', Y', Z')$ where

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = R(\theta_y, \theta_x, \theta_z) \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \quad (3.5)$$

As f is the distance from camera optical center to the center of each image, the perspective projection of a 3D point onto an image plane is given by

$$x = \frac{f}{Z} \cdot X,$$

$$y = \frac{f}{Z} \cdot Y.$$

Denoting

$$V = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

we can rewrite the coordinates of the image point in a matrix form as

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \propto V \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (3.6)$$

where $\propto$ means proportional to within a scale factor. The same holds for the image point in the second image $p' = (x', y')$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \propto V \cdot \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix}.$$

Substituting Equation (3.5) into the above Equation yields

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \propto V \cdot R(\theta_y, \theta_x, \theta_z) \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}.$$

Combining this with Equation (3.6) gives

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \propto V \cdot R(\theta_y, \theta_x, \theta_z) \cdot V^{-1} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

Thus, the perspective transformation matrix we seek has the following special form

$$M = V \cdot R(\theta_y, \theta_x, \theta_z) \cdot V^{-1}. \tag{3.7}$$

Let an image point $(x, y, 1)$ be mapped by M to $(\tilde{x}, \tilde{y}, 1)$. Our goal is to determine the above unknowns so that the *matching error*,

$$E = \sum_{i=0}^{n} \| (x'_i, y'_i) - (\tilde{x}_i, \tilde{y}_i) \|^2,$$

is minimized. This is a nonlinear problem that does not have a closed form solution. We have devised a method to solve it using an iterative technique, in which each iteration entails solving a linear equation, as we now explain.

3.5.2 Initial Parameter Estimation

First, the focal length is estimated using an 8-parameter model—we represent the transformation using a general perspective transformation, whose matrix is arbitrary up to an overall scale factor. Four matching pairs of points are chosen which are well separated in both x and y, and are far from being collinear. A linear system in the homogeneous coordinates of these four points can be set up as

$$\mathbf{x}'_i = M\mathbf{x}_i \qquad \text{for } i = 1, \ldots, 4 \tag{3.8}$$

and solved to find the perspective transformation M using Equ.(3.3). From M, f is estimated as explained below. Although this process is not accurate, the focal length does not have to be found accurately at this stage, as the subsequent calculations are relatively insensitive to errors in the estimate for f (see Section 5.3.3), which justifies the simple approach above based on only four matching points. The estimate for f is refined at a later stage of panorama construction.

To find f, we expand the rotation matrix R as

$$R = V^{-1} \cdot M \cdot V = \begin{pmatrix} m_0 & m_1 & m_2/f \\ m_3 & m_4 & m_5/f \\ m_6 f & m_7 f & 1 \end{pmatrix}.$$

If M has exactly the desired special form, R is a rotation, and any two rows or columns should be orthogonal and each row or column vector should have unit magnitude. In practice, matching and numerical errors may prevent these conditions from being exactly satisfied. However, by assuming that they are, we can estimate f in various ways. Following [Shum00], equality of magnitude of the first two rows, and orthogonality of the first two rows, leads to the following estimates for f^2:

$$f^2 = \frac{m_5^2 - m_2^2}{m_0^2 + m_1^2 - m_3^2 - m_4^2},$$

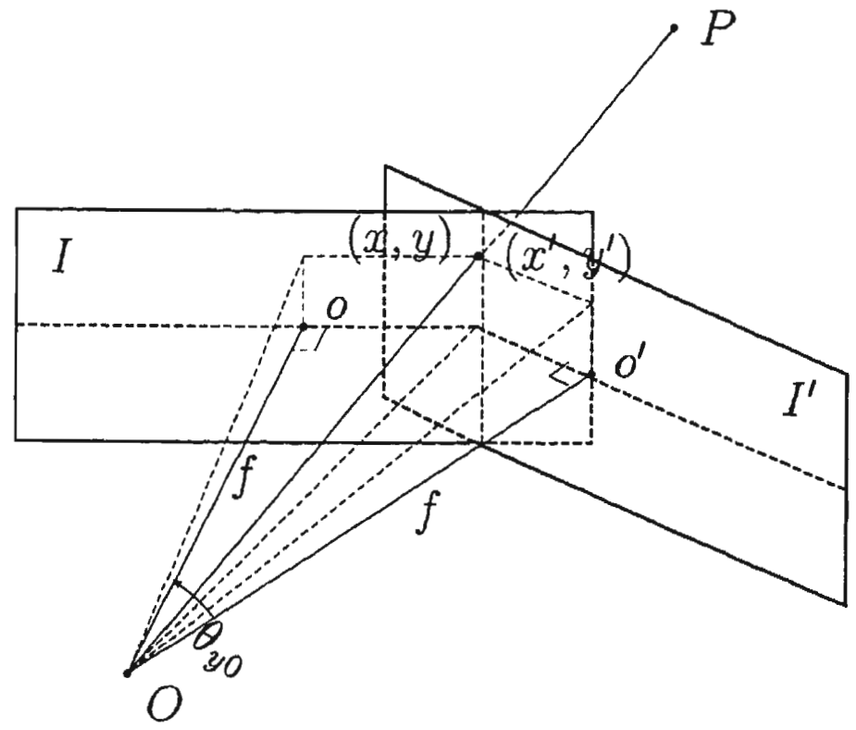


Figure 3.12: Estimation of initial angles

$$f^2 = \frac{m_2 m_5}{-m_0 m_3 - m_1 m_4}.$$

The initial estimate for focal length is then taken to be the average of these two estimates.

Let θ_{y0}, θ_{x0}, and θ_{z0} be the initial estimates of pan, tilt, and roll angles. We assume that $\theta_{z0} = 0$, since the user is much less likely to roll the camera significantly than to tilt it, because there is often a horizon or other horizontal reference line in the picture. Initial pan and tilt angles are estimated by averaging the pan and tilt angles over all tie-points:

$$\theta_{x0} = \sum_{i=0}^{n} (\arctan \frac{y_i'}{f} - \arctan \frac{y_i}{f})/n,$$

$$\theta_{y0} = \sum_{i=0}^{n} (\arctan \frac{x_i'}{f} - \arctan \frac{x_i}{f})/n,$$

respectively. See Figure 3.12.

3.5.3 Accurate Parameter Determination

Given the above estimates for f, θ_y, θ_x, and θ_z, we can construct M_0, our initial estimate of the transformation matrix, using Equation 3.7.

Note that M_0 should not be computed from Equations (3.8) and (3.3) as it is likely to be insufficiently accurate for image alignment purposes, in that it may differ too much from the required special form based on rotations, due to the extra free parameters in the general model, and lack of precision in locating feature points, see an example in next chapter, Figure 4.6.

The initial estimate M_0 is refined by finding subsequent approximations M_k by means of incremental updates. More specifically, let M_k be the current approximation. An update matrix ΔM_k is found by minimizing the error E induced by M_k. Then the subsequent approximation M_{k+1} to the correct M is given by

$$\begin{aligned} M_{k+1} &= \Delta M_k \cdot M_k \\ &= \Delta M_k \cdot \Delta M_{k-1} \cdots \Delta M_0 \cdot M_0. \end{aligned}$$

We explain how to determine ΔM_k below.

Suppose that an approximate matrix M_k has already been found in the form

$$M_k = V R_k V^{-1}.$$

Let us consider applying an incremental rotation matrix ΔR_k with angles $\Delta\theta_x$, $\Delta\theta_y$, and $\Delta\theta_z$ to the rotation part of M_k to produce the next approximation M_{k+1}. Thus

$$\begin{aligned} M_{k+1} &= V \cdot \Delta R_k \cdot R_k \cdot V^{-1} \\ &= V \cdot \Delta R_k \cdot V^{-1} \cdot V \cdot R_k \cdot V^{-1} \\ &= \Delta M_k \cdot M_k, \end{aligned} \tag{3.9}$$

where

$$\Delta M_k = V \cdot \Delta R_k \cdot V^{-1}, \tag{3.10}$$

and, using the notation of Equation (3.4),

$$\Delta R_k = R(\Delta\theta_y, \Delta\theta_x, \Delta\theta_z). \tag{3.11}$$

Let $d_x = \Delta\theta_x$, $d_y = \Delta\theta_y$, and $d_z = \Delta\theta_z$. Because the incremental angles are small, and $\sin\theta \approx \theta$ and $\cos\theta \approx 1$ for small θ, it follows that

$$\begin{aligned} \Delta R_k &\approx \begin{pmatrix} 1 & 0 & d_y \\ 0 & 1 & 0 \\ -d_y & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -d_x \\ 0 & d_x & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & -d_z & 0 \\ d_z & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & -d_z & d_y \\ d_z & 1 & -d_x \\ -d_y & d_x & 1 \end{pmatrix}. \end{aligned} \tag{3.12}$$

Combining (3.10) and (3.12) yields

$$\Delta M_k \approx \begin{pmatrix} 1 & -d_z & d_y f \\ d_z & 1 & -d_x f \\ -d_y/f & d_x/f & 1 \end{pmatrix}. \tag{3.13}$$

Since the focal length in pixel units is normally much larger than the magnitudes of d_y and d_x, we may assume that the last row of ΔM_k is approximately $(0, 0, 1)$. This observation allows us to formulate a *linear* optimization problem for the variables d_x, d_y and d_z, which makes the problem tractable.

Suppose that $(x^k, y^k, 1)$ is mapped to $(x^{k+1}, y^{k+1}, 1)$ by ΔM_k:

$$\begin{pmatrix} x^{k+1} \\ y^{k+1} \\ 1 \end{pmatrix} = \Delta M_k \begin{pmatrix} x^k \\ y^k \\ 1 \end{pmatrix}$$

After the update by ΔM_k, the matching error becomes

$$E = \sum_{i=0}^{n} \| (x_i', \quad y_i') - (x_i^{k+1}, \quad y_i^{k+1}) \|^2$$

$$\approx \sum_{i=0}^{n} \left| \begin{pmatrix} x_i' \\ y_i' \end{pmatrix} - \begin{pmatrix} 1 & -d_z & f d_y \\ d_z & 1 & -f d_x \end{pmatrix} \begin{pmatrix} x_i^k \\ y_i^k \\ 1 \end{pmatrix} \right|^2$$

$$= \sum_{i=0}^{n} \left| \begin{pmatrix} \Delta x_i^k \\ \Delta y_i^k \end{pmatrix} - \begin{pmatrix} 0 & -d_z & f d_y \\ d_z & 0 & -f d_x \end{pmatrix} \begin{pmatrix} x_i^k \\ y_i^k \\ 1 \end{pmatrix} \right|^2$$

where $\Delta x_i^k = x_i' - x_i^k$, $\Delta y_i^k = y_i' - y_i^k$.

Minimizing of the matching error is achieved by setting partial derivatives of E with respect to d_x, d_y, and d_z to zero. This gives

$$\begin{cases} \sum_{i=0}^{n} (\Delta y_i^k - d_z x_i^k + d_x) = 0 \\ \sum_{i=0}^{n} (\Delta x_i^k + d_z y_i^k - d_y) = 0 \\ \sum_{i=0}^{n} [(\Delta x_i^k + d_z y_i^k - d_y) y_i^k - (\Delta y_i^k - d_z x_i^k + d_x) x_i^k] = 0 \end{cases},$$

which can be written as

$$A \cdot (d_x, d_y, d_z)^T = B \tag{3.14}$$

where

$$A = \begin{pmatrix} nf & 0 & -\sum x_i^k \\ 0 & nf & -\sum y_i^k \\ \sum x_i^k f & \sum y_i^k f & -\sum ((y_i^k)^2 + (x_i^k)^2) \end{pmatrix},$$

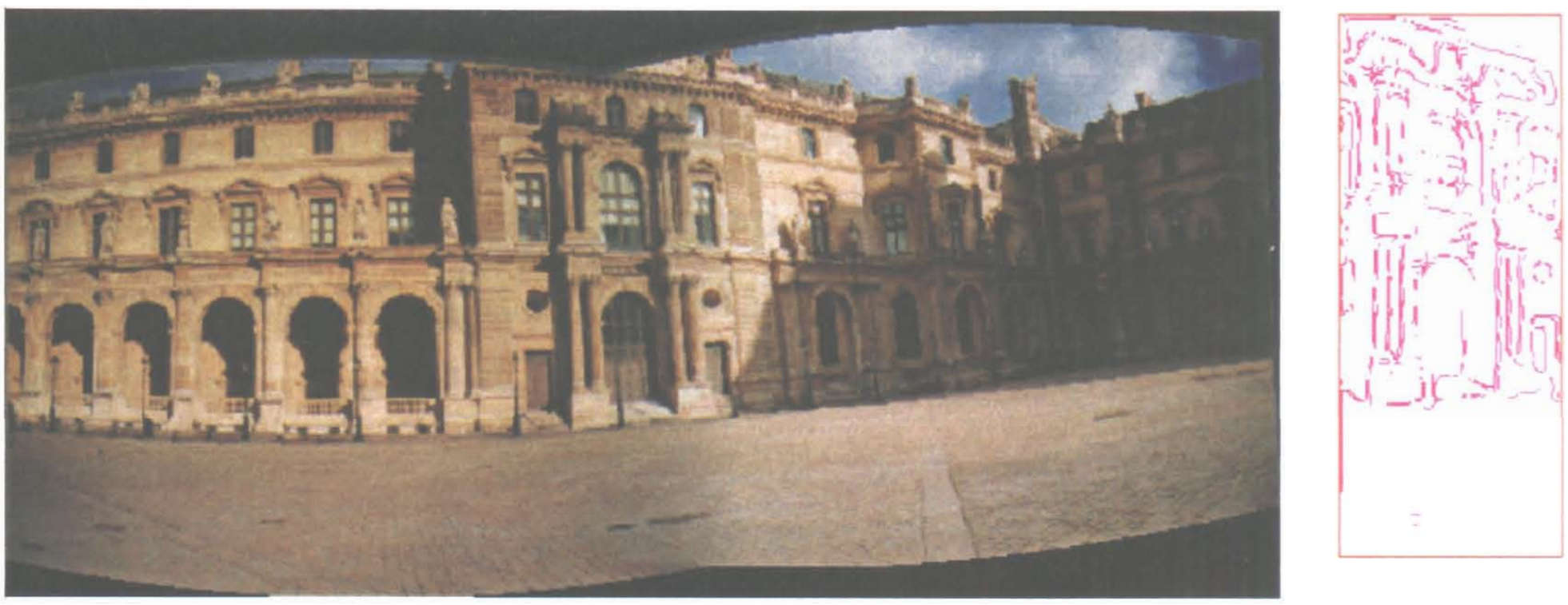
$$B = \left(-\sum \Delta y_i^k, \sum \Delta x_i^k, \sum (\Delta x_i^k y_i^k - \Delta y_i^k x_i^k) \right)^T.$$

These linear equations can easily be solved for the angular increments (d_x, d_y, d_z). The incremental update matrix ΔM_k is then given by (3.10).

By using the original matrix ΔM_k in Equation (3.10), instead of its approximation in (3.13), for updating, we ensure that the rotational constraints are preserved: otherwise, errors may accumulate.

Our experiments show that the series M_k converges rapidly and 4 to 5 updates are normally sufficient for E to reach a small value.

An example of the registration produced using the above approach (after mapping the images onto a cylinder) is shown in Figure 3.13(a).



(a) (b)

Figure 3.13: Registered image and difference on edge image in overlapped region

The right part of Figure (b) is the difference on edge image in overlapped region.

3.6 Experiments

In this section, we provide experimental demonstrations of the effectiveness of the feature-based registration algorithm presented in this chapter. A variety of images, deriving from differing sources, have been tested: an analogue film camera, a digital camera, and computer generated images.

To choose enough features, we have found in practice that the parameters need to be set as follows: $\sigma = 5$ at most for the standard deviation of the Canny edge detector, $T_g = 0.3$ for the threshold value for converting grayscale edge images to binary images, and $l = 20$ with $C = \frac{\pi}{4l}$ for maximum curvature point detection. If they are given lower values, more features are found, and the algorithm takes longer but the final results are only marginally improved. Thus we recommend that for most users, the parameters be set to $\sigma = 2.5$, $T_g = 0.2$ and $l = 10$. We recommend that the matching window size be set to $s = 30$, i.e. the template used be 30×30 pixels. We also recommend that a pair of salient points whose gradient metric value is greater than $T_1 = 0.6$ be taken as matched candidates, those less than $T_2 = 0.3$ be considered insignificant and not used for image registration, and those between T_1 and T_2 be put into the candidate list if accepted by the shape metric. The clustering threshold is set to $d = 0.035$. The user can change some of these values, but in practice rarely needs to do so. For all the examples shown below in this chapter, we set the parameters to be the same as the above recommended values.

Two sample images scanned from film photographs were already shown in Figure 3.1. The images have both intensity differences and large relative perspective distortions. Detected tie-points (corresponding feature points) are marked '+' in Figure 3.10. Thirteen tie-points were detected. The resulting composite image after registration, mapping onto a cylindrical surface and developing, was shown in Figure 3.13. In this example, the minimum image overlap was 16%.

Normally, for building a cylindrical panorama, we expect the tilt rotation to be small, but our registration method does allow recovery of a large tilting angle as well as panning angle, if necessary, as seen in Figures 3.14–3.16. Seven tie-points were found in Figure 3.15; The feature extracting region of the left image is 16%,

Figure 3.14: Images with perspective distortion and a large panning as well as tilting.

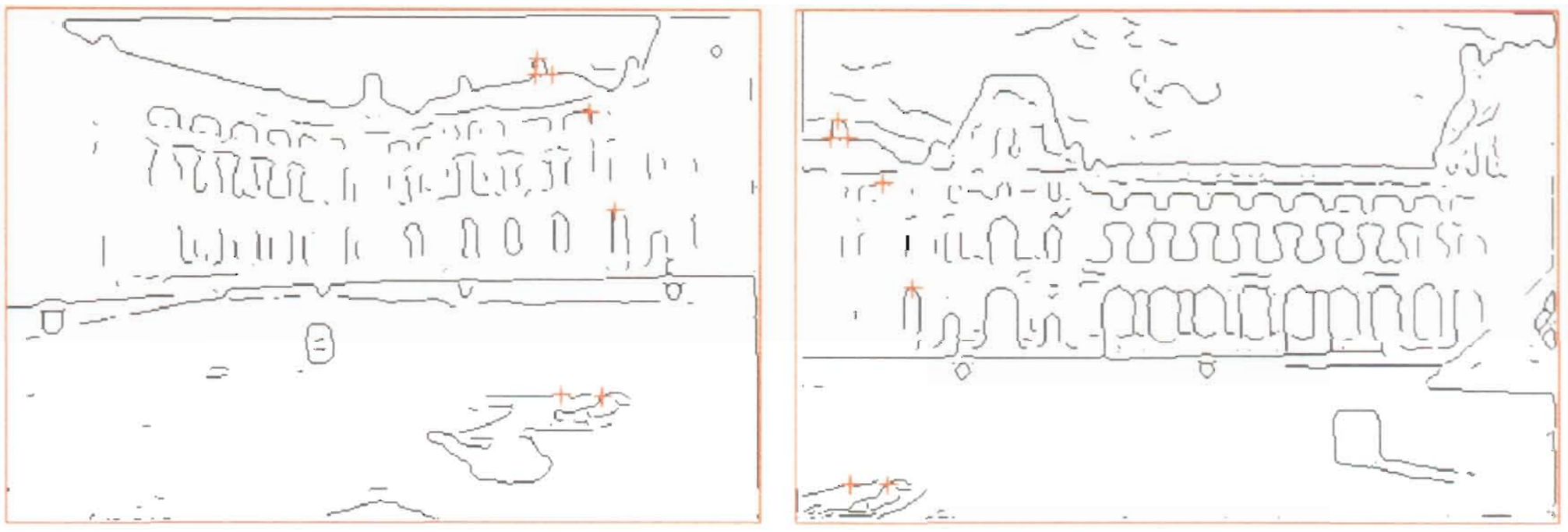


Figure 3.15: Tie-points identified in each image

which is the default minimal overlap region. The result of registration is shown in Figure 3.16. Again, this pair of images came from scanned film photographs.

To show the robustness of the algorithm, we include an example of richly textured images: see Figure 3.17. The images in this case were captured using a digital camera. Since our method takes note of the image structure in the overlapping region, it can easily align these two images with each other. Twenty two tie-points were identified, shown in Figure 3.18. The aligned image is shown in Figure 3.19. Another example of registering highly texture images are shown in Figure 3.20, 3.21 and 3.22, which are scanned in film photographs remotely taken.

Figure 3.16: Registered image (building)

Figure 3.17: Images with heavy texture

An example of computer generated images is shown in Figures 3.23. When the number of features extracted on the minimum overlap region of the left image is not sufficient (less than 6, for example), the region is extended in order to include more features, such as it was extended to 50% in this example. Thirteen pairs of tie-points were found—see Figure 3.24. The aligned images after transformation are shown in Figure 3.25.

In Table 3.2 we list the estimated initial parameters and the final transformation matrix obtained from optimization. The initial parameters are pan and tilt

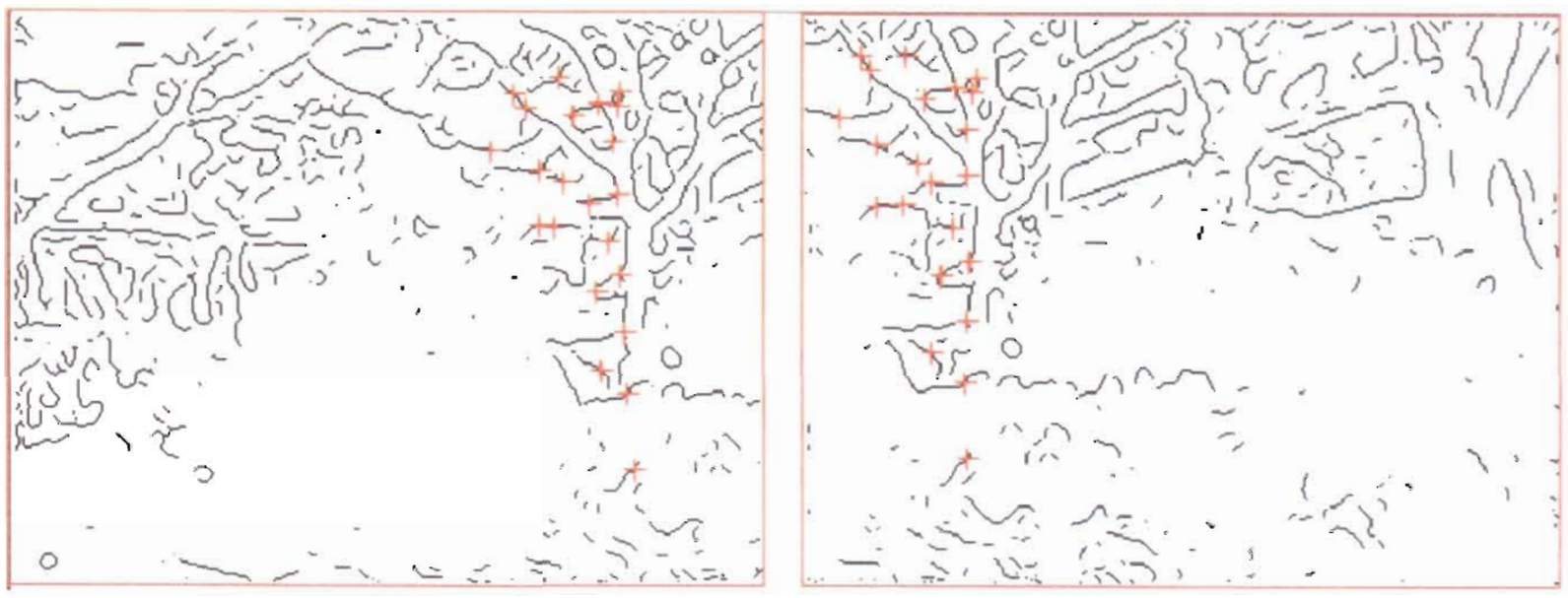


Figure 3.18: Tie-points identified on edges

Figure 3.19: Registered image(tree)

angles, focal length, and the initial transformation matrix calculated from these values. The final transformation parameters for aligning each image pair shown in Figures 3.1, 3.14, 3.17 and 3.23 are listed below each of their initial parameters. Note that here we show the final result of 8 parameters of transformation matrix but not the final pan, tilt and roll angles. That is because we use the original matrix ΔM_k in Equation (3.10) for updating each step in the optimization process, where it is tricky to reversely calculate the angles from the rotation matrix in Equation 3.7, though using the angles to show the result would be better for comparison.

Images	θ_{y0}	θ_{x0}	f	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
Fig.3.1	−0.777	0.005	334.0	0.713	-0.003	-231.577	0.000	1.000	-1.629	0.002	0.000	0.713
aligned				0.704	0.067	−234.720	−0.074	1.000	6.443	0.002	0.000	0.704
Fig.3.14	−0.735	0.120	334.0	0.743	-0.0791	-219.616	0.000	0.993	-39.051	0.002	0.000	0.738
aligned				0.743	−0.012	−222.71	−0.070	0.997	−31.604	0.002	0.000	0.733
Fig.3.17	−0.514	−0.018	358.0	0.871	0.009	-175.96	0.000	1.000	6.325	0.001	0.000	0.871
aligned				0.868	0.059	−177.42	−0.052	1.000	10.465	0.001	0.000	0.868
Fig.3.23	−0.260	0.002	259.8	0.967	-0.001	-66.036	0.000	1.000	-1.480	0.001	0.000	0.967
aligned	−0.260	0.002	259.8	0.966	0.009	−66.90	−0.009	1.000	0.086	0.001	0.000	0.966

Table 3.1: Estimation of parameters

The test results of this feature-based registration approach on 10 sets of image sequences, total 122 pairs of images, are 76% visually fine registered, 21% coarsely registered, 3% failed. See Table 3.1, the first column is the name of image sequence, the second, third, and fourth columns are the number of image pairs that are finely registered, coarsely registered, and not registered respectively. The fifth column is total number of image pairs. The last column shows the image type (whether they are taken from film camera or digital camera). The last two rows in the Table are the total number and percentage of each group.

The algorithm presented in this chapter is quite robust and reliable as long as corresponding features are available. If insufficient feature information is extracted, we only keep the initial parameter estimate but do not performing the feature based parameter optimization. Further parameter refinement is done by a gradient based registration method given in the next chapter.

If we carefully examine Figure 3.16, we can observe some blur in the middle of the aligned image, suggesting impreciseness of the estimated transformation parameters. Assuming that this error is not due to camera translation, the parameter errors may be due to two causes: one is error in locating feature points, which generally has a small effect on the result. The other is due to mismatched features, which can have a very adverse effect on the result. If the latter case happens, it is

Image sequence	Fine	Coarse	Fail	Total pairs	Type
lo	7	1	0	8	Scanned in film photos
mBld	3	4	1	8	Scanned in film photos
hku	10	4	2	16	Scanned in film photos
japl	8	4	0	12	Scanned in film photos
mbl	7	2	0	9	digital camera photos
mbm	8	5	0	13	digital camera photos
scysl	15	0	0	15	digital camera photos
cycl	13	2	0	15	digital camera photos
peak	9	2	1	12	digital camera photos
garden	12	2	0	14	digital camera photos
Sum	92	26	4	122	
Percentage	0.76	0.21	0.03		

Table 3.2: Results of feature-based registration

likely that a large residual error will remain from the optimization procedure. In such a situation, the result of optimization is discarded as unreliable, and the initial parameter estimates are kept to be used as the starting point for the gradient based fine registration.

Tests show that our method is sufficiently fast for realistic use in a commercial application. This feature-based registration method running on a Pentium III 500 MHz PC takes about 2∼3 seconds for a typical image pair shown in this chapter, with image size ranging between 320∼390 pixels in width and 240∼260 in height. Canny edge detection takes about 1.5 seconds out of this time. Since there are only three rotation parameters involved in the optimization procedure, a small set of corresponding pairs can achieve a visually good result. This makes both the matching process and the approximation of the transformation computationally efficient.

3.7 Summary

In this chapter, we have shown how images taken from a common viewpoint can be registered using a feature matching scheme. The approach results in a fast method that produces visually good results when a large panning rotation as well as small tilting and rolling has occurred between adjacent images. In the first stage of our method, salient feature points of maximum curvature on edges in the images are located. The maximum curvature point detector is set up so that it finds a small number of salient feature points in each image. In the second stage, feature points in the right image are matched with feature points in the left image. Two matching metrics are used, a gradient metric and a shape metric; a preprocessing step is used to quickly reject impossible matches. The motivation here is to tackle

the difficulties of exposure and lighting differences in image acquisition, as well as the relative perspective distortions between the images. The gradient metric used is particularly robust against brightness variations between the two images. The use of a large deviation factor in edge detection not only smoothes out image noise but also helps to overcome distortions. The shape metric is designed to further pick up correspondences under relatively large perspective distortions. In the last stage, a perspective transformation is determined by an approximation procedure using the matched features. We have shown how this optimization problem can be realized by an iterative scheme with linear steps, where the transformation based on our application of the rotation model is represented with four unknown parameters. To obtain the starting parameters for performing the optimization, initial camera pan and tilt angles are estimated from the matched features assuming a zero initial camera rolling.

Since the 4-parameter feature-based model includes all the camera constraints in our case, a small number of tie points are sufficient to provide reasonably good overall alignment. Our method is adequately fast. Canny edge detection is the step which overall takes the most time.

The method works well as long as a sufficient number of features are extracted in the overlap region and similar features are not repeated too often over the whole region. Otherwise, the approach may only provide a coarse alignment. To improve the registration resolution, we may further invoke a gradient-based fine registration method to be described in the next chapter. This is done if the number of correspondences is too small, or the residual error from optimization process used to estimate the transformation parameters exceeds a threshold.

The proposed algorithm serves as an important building block for the whole system of panorama construction.

Figure 3.20: Remote imaging

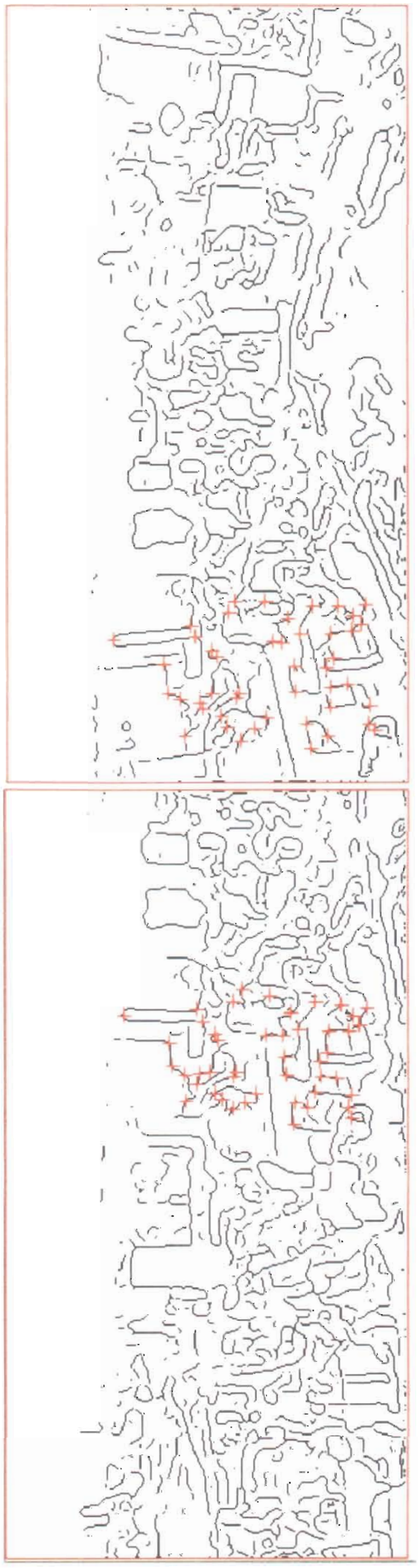


Figure 3.21: Tie-points identified on edges

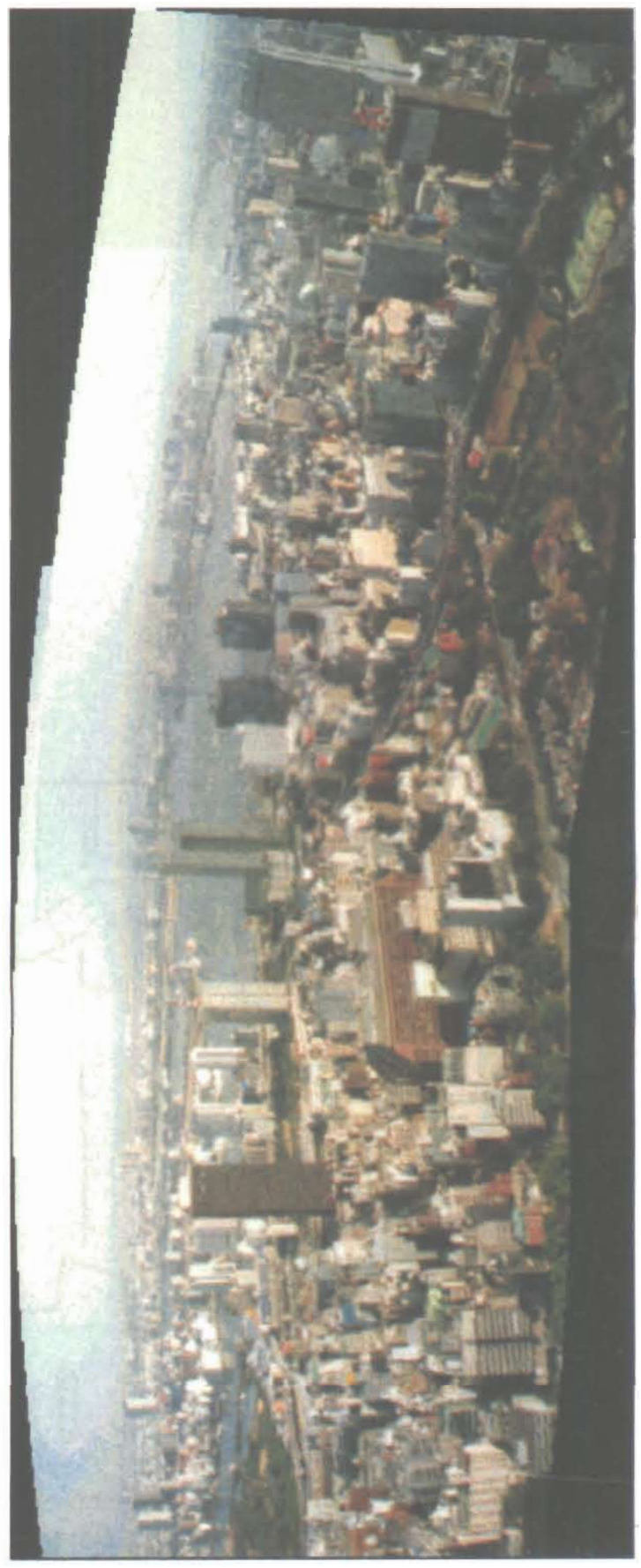


Figure 3.22: Registered image

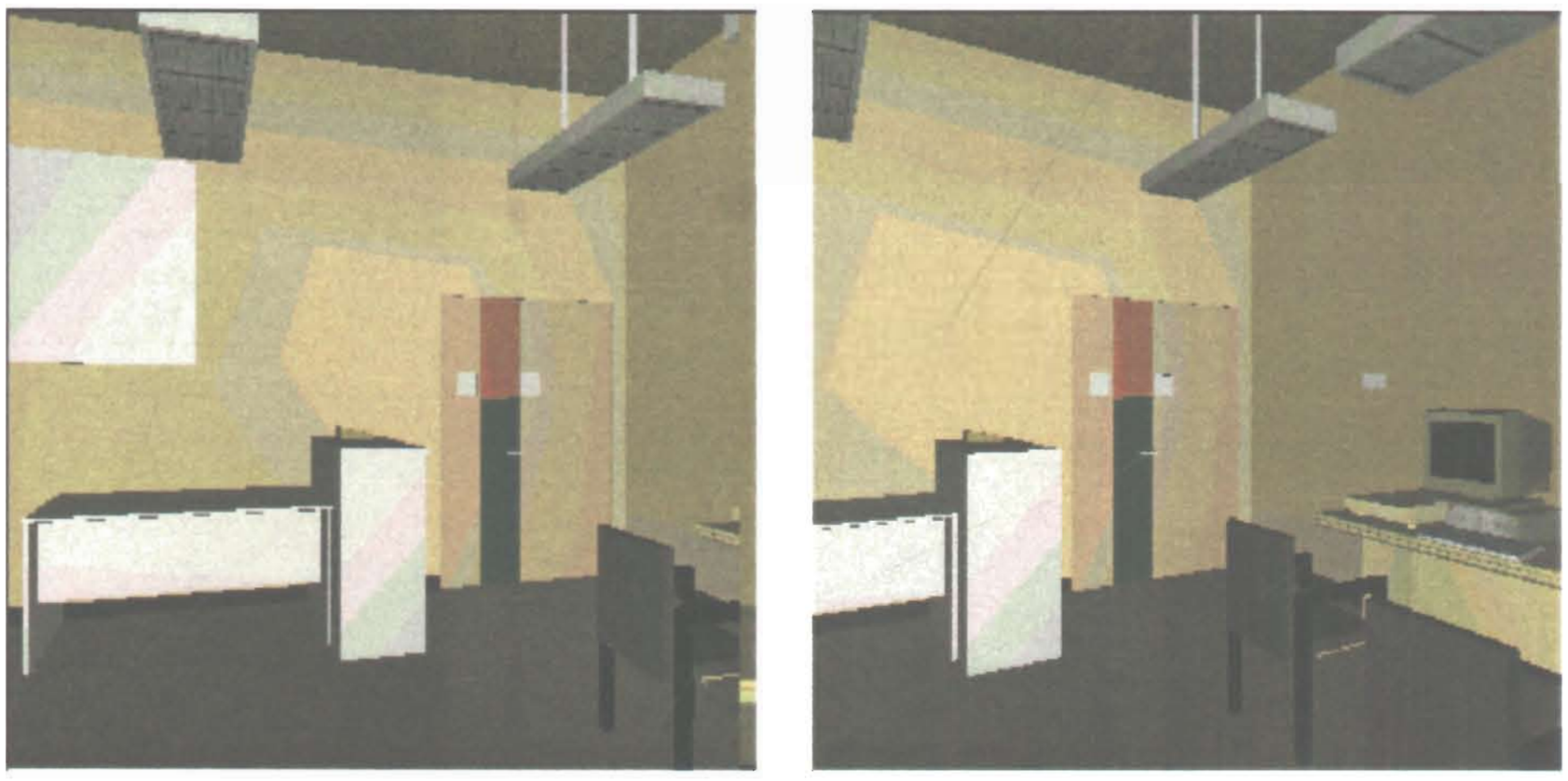


Figure 3.23: Images generated by computer

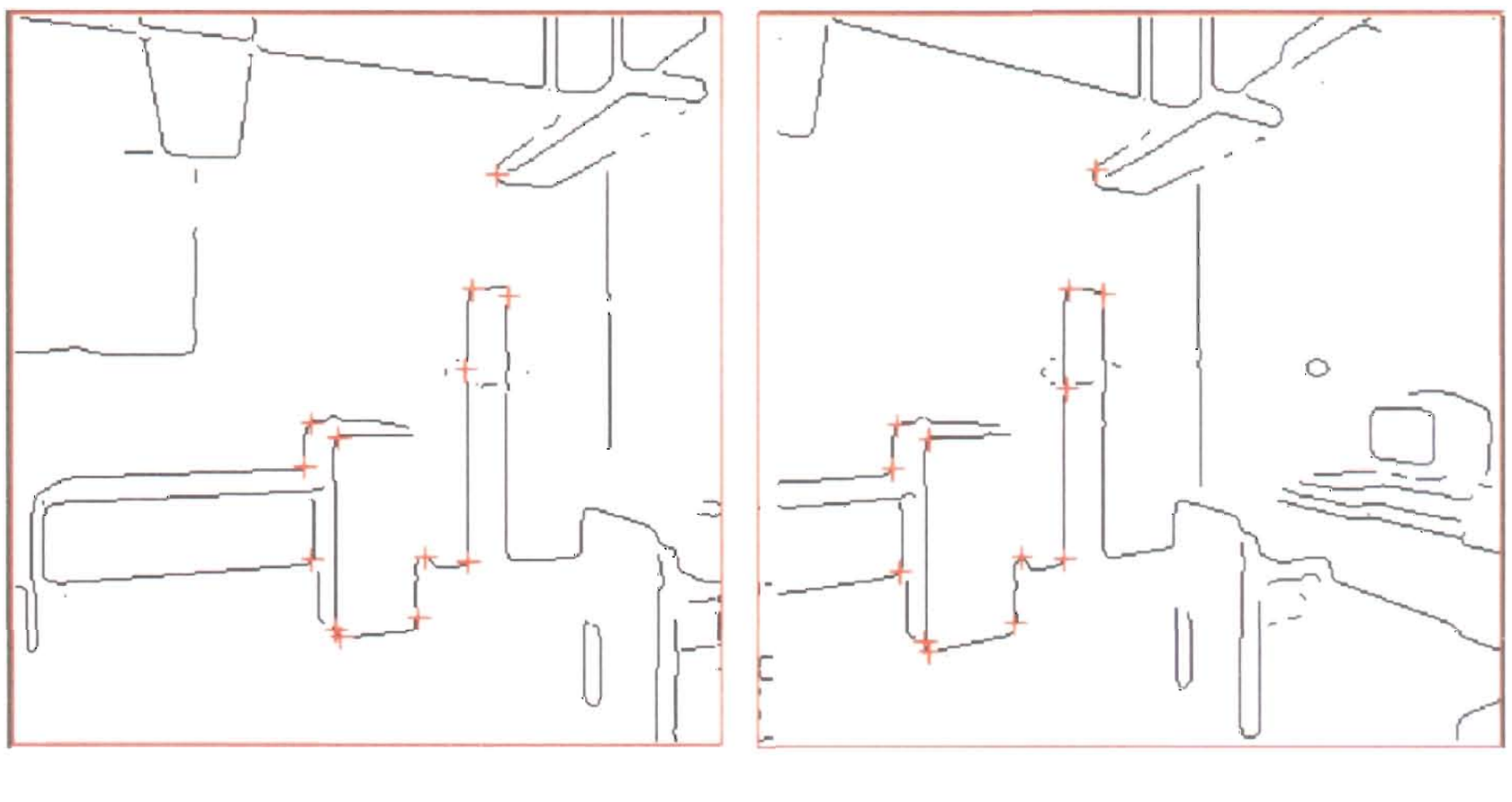


Figure 3.24: Tie-points identified on edges

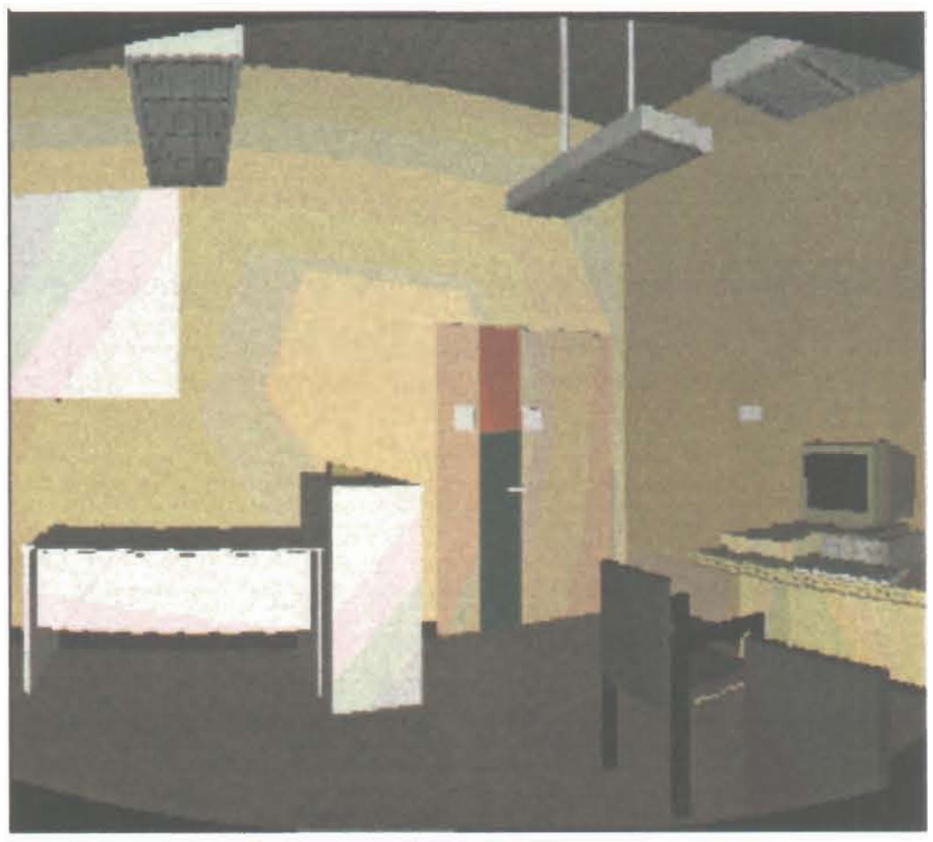


Figure 3.25: Registered image

Chapter 4

Fine Registration

This chapter provides a study of the application of gradient-based motion detection techniques (i.e. optical flow methods) to registration of adjacent images taken using a hand held camera for the purposes of building a panorama. A general 8-parameter model and a more compact 3-parameter model for transformation estimation are described from two different starting points. Both of these models are approximations of the real situation when the viewpoint position is not absolutely fixed but includes a small translation, and thus distortion and blurring are sometimes present in the final registration results. We propose a new 5-parameter model that shows better results and has less strict requirements on good choice of initial unknown parameters. An analysis of the displacement recovery range and its enlargement using Gaussian filters are also given.

4.1 Overview

Image registration can also be viewed as camera motion recovery. The problem of recovering motion information from optical flow has been extensively studied for

two decades [Alvarez00, Aubert99, Barron94, Bergen92, Horn81, Nage86, Verri89]. In this chapter, we review the basic idea of the technique and show how to use it to solve our particular problem of registering partially overlapped images taken from an approximately fixed viewpoint.

Optical flow is the apparent motion of intensity patterns when the objects that give rise to them move, or equivalently, a moving camera is pointed at a static scene. In simple situations, optical flow is the projection of the velocity of moving 3D objects onto the 2D image plane. It is described in terms of spatial and temporal derivatives of intensity in the image, and it can be used to recover object or camera motion and object surface shape. Optical flow based methods for motion recovery are also referred to as *gradient based* methods. It is a classical method for recovering continuous object motion. Since in our application, the two adjacent images are already roughly aligned using the feature-based method presented in the previous chapter, their sub-images in the overlap region can be regarded as having small difference due to camera motion. Thus, the fine registration problem now is subject to the approach of optical flow.

Optical flow approaches are different from feature-based methods discussed in the previous chapter. The latter are discrete methods in the sense that only feature points are considered, while the optical flow method may be viewed as a continuous method in that all image points contribute to the calculation. It is a full density approach in that it can recover the motion of every pixel in the image. The disadvantage of optical flow methods is that they usually assume a small change between neighboring images (i.e., the scene is displaced by a few pixels at most), while feature based methods generally do not require that assumption.

For a fully automatic approach to registration, we thus first use the feature-based method presented in the last chapter to coarsely align the images to within

a few pixels, and then employ a gradient-based method to refine the registration, using much denser information.

The image alignment problem we consider in panorama building is restricted to a special type of motion that enables the optical flow method to give meaningful solutions. The motion model is termed a planar surface flow model, or a rigid body model in [Bergen92], or a 3-parameter model in [SzelS97]. We discuss these models in more detail shortly. A new 5-parameter model is also proposed here.

Optical flow methods are based on the assumption of smoothly varying intensities in an image, whereas real images have sharp edges. We thus employ Gaussian smoothing to improve the performance of our optical flow method, which in particular allows us to handle large displacements in a different manner to the traditional hierarchical coarse-fine strategy. An analysis of the relationship between the smoothing factor and the displacement which can be recovered is given.

There are two steps in optical flow based methods: the first is estimating optical flow, the second is estimating camera motion from the optical flow. In Section 4.2 we introduce basic formulae for optical flow. Section 4.3 gives the particular form of optical flow for a moving planar surface, while Section 4.4 derives the same result from the viewpoint of a planar perspective transformation. Discussion of the models and test results are given in Section 4.5. The use of image smoothing to allow registration with large image displacements is studied in Section 4.6. We summary these ideas in Section 4.7.

4.2 Optical Flow Approach

When an object moves in front of a camera, or a camera moves through a static environment, there are corresponding changes in the images taken with the camera.

These changes provide information that allows us to recover the relative motion as well as shape information about the objects. In this section, we begin by reviewing mathematical descriptions for the brightness changes in images caused by motion, giving a definition of optical flow and the basic constraint it provides. Then, in particular, we consider motion models which can be described by a linearized optical flow constraint; which allows a least squares solution to motion recovery and hence image registration.

4.2.1 Definition

Let p be the image point of a 3D object point. Then either moving the object or the camera will cause a movement of the image point. Let (x, y) denote the coordinates of p in the image. Then the velocity of its movement in the image is

$$u = \frac{dx}{dt}, \quad v = \frac{dy}{dt}.$$

$\mathbf{u} = (u, v)^T$ is the vector field of 2D velocities, commonly known as the *optical flow* field.

The optical flow at each point in the image is the instantaneous velocity of the brightness pattern at that point. In particularly simple situations, such as a moving planar surface, the apparent velocity of the brightness patterns can be directly related to the movement of surfaces in the scene or the motion of camera. Computing the motion of points on the object or camera, and hence registering successive images is a matter of simple geometry once the optical flow is known. Thus, first we need to estimate optical flow. We give below a description of the basic constraint formula of optical flow [Horn81].

4.2.2 Basic formula

At time t, let $I(x, y, t)$ be the intensity at image point (x, y). This point is the projection of an object point (X, Y, Z) onto the image plane at time t. At time $t+\delta t$, suppose this object point moves to $(X+\delta X, Y+\delta Y, Z+\delta Z)$. Its projection point onto the image plane moves to $(x+\delta x, y+\delta y)$, and the intensity at this point is $I(x+\delta x, y+\delta y, t+\delta t)$. A basic assumption of the optical flow approach is that this intensity is the same as it was at time t, that is

$$I(x+\delta x, y+\delta y, t+\delta t) = I(x, y, t)$$

If the object is smooth and moves smoothly, i.e. the brightness varies smoothly with respect to x, y and t, we can expand the left-hand side of the equation above in a Taylor series. Simplifying and omitting the second and higher-order terms, we obtain

$$I(x, y, t) + \delta x \frac{\partial I}{\partial x} + \delta y \frac{\partial I}{\partial y} + \delta t \frac{\partial I}{\partial t} = I(x, y, t)$$

Thus in the limit as $\delta t \to 0$, we have

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

i.e.

$$u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0, \tag{4.1}$$

where $u = \frac{\partial x}{\partial t}, \quad v = \frac{\partial y}{\partial t}$. This is called the optical flow constraint equation, image brightness constraint, or gradient constraint, and is the basic formula for estimation of optical flow [Horn81, Anandan89]. The above equation can also be written as

$$(\nabla I)^T \mathbf{u} = -\Delta I \tag{4.2}$$

where $\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})^T$ is the spatial image gradient, and $\Delta I = \frac{\partial I}{\partial t}$ is the temporal image intensity derivative of identical points in different frames.

An important issue when using the basic constraint formula is its reliability. Notice that the intensity conservation assumption implies that the image intensity associated with the projection of a 3D point does not change as the object or the camera moves. This assumption is only approximately true in real image sequences, and ignores possible changes in intensity due to variation in illumination. For example, Verri and Poggio [Verri89] have performed a careful study of the problem considering various factors such as lighting, reflection and texture. Their conclusion is that while the basic formula is a necessary and sufficient condition for optical flow to be equal to the projection of 3D velocity onto the image plane, except in a few special cases, the basic formula is not a reliable or accurate approximation. So, generally, estimates of motion and surface from optical flow are not reliable. Fortunately, our aim is to register photos taken from a nearly fixed viewpoint, and this special problem *is* one of the cases to which the optical flow method can be reliably applied. The approach is also often referred as *gradient-based* registration method.

The optical flow constraint equation provides only one equation but there are two unknown variables u and v. So more constraints must be introduced to find a general solution for optical flow. These extra constraints may be smoothness constraints on optical flow or on image gradient [Horn81, Terzopoulos86, Nage87]. Under these smoothness assumptions, iterative schemes have been developed for solving the problem. We omit discussion of these further constraints here. Instead, we will see later that the basic constraint equation (4.1) is sufficient to solve our particular problem of panorama image stitching. Nevertheless, most optical flow methods are based on minimizing an error value

$$E_c = \int\int ((\nabla I)^T \mathbf{u} + \Delta I)^2 dxdy, \tag{4.3}$$

the total amount by which the basic formula is not satisfied, calculated over a

suitable region across multiple images.

Note that in the above error measure, if the optical flow components u and v can be represented as linear functions of some motion parameters which is true in some particular cases, the error term is quadratic in terms of these parameters. In such a case, a least-squares solution that leads to a linear system can be employed to determine the parameter values.

4.2.3 Least Square Solution

Taking the above idea further, let us consider a particular transformation model where the optical flow field is linear in terms of some motion parameters. We show that this is indeed the case for pure camera rotation situations in Sections 4.3 and 4.4. Other general models are beyond the scope of this research. In this case, we may write

$$\mathbf{u} = \mathbf{g}(\mathbf{x})\mathbf{d} \tag{4.4}$$

where $\mathbf{g}(\mathbf{x})$ is a coefficient matrix, saying how $\mathbf{u}$ varies with x and y and depends on the parameters:

$$\mathbf{g}(\mathbf{x}) = \begin{pmatrix} g_{00}(\mathbf{x}) & \dots & g_{0n}(\mathbf{x}) \\ g_{10}(\mathbf{x}) & \dots & g_{1n}(\mathbf{x}) \end{pmatrix},$$

and $\mathbf{d} = (d_0, d_1, ...d_n)^T$ is the parameter vector.

In the above, the $g_{ij}(\mathbf{x})$ depends on pixel's position and camera calibration parameters, while the d_i could be a motion parameter such as camera rotation velocity or camera translation velocity, or a combination of motion and surface parameter. We will introduce various ways of representing $g_{ij}(\mathbf{x})$ and d_i for each particular transformation model in the next two sections. Here the general representation is used for obtaining a solution in general.

Taking a pair of successive images to be one unit of time apart, we may denote the error in intensity (i.e., amount by which the constraint equation is not satisfied) at a point $\mathbf{x}$ to be

$$\Delta I(\mathbf{x}) = I(\mathbf{x} + \mathbf{u}_i, \mathbf{t} + 1) - I(\mathbf{x}, \mathbf{t}) \tag{4.5}$$

Our objective is to minimize the total error in the optical constraint Equation(4.2) over a suitable region of the image. Hence, the two images(portions) are aligned. The integral in Equation (4.3) is now taken over summation over a suitable set of pixels $(x_i, y_i) \in \mathbf{x}$. Hence

$$\begin{aligned} E_c(\mathbf{x}) &= \sum_{\mathbf{x}} ((\nabla I)^T \mathbf{u} + \Delta I)^2 \\ &= \sum_{\mathbf{x}} ((\nabla I)^T \mathbf{g}(\mathbf{x})\mathbf{d} + \Delta I)^2 \end{aligned} \tag{4.6}$$

should be minimized.

Minimizing the above error function with respect to $\mathbf{d}$ requires

$$\frac{\partial E_c}{\partial d_i} = 0, \quad i = 0...n$$

and hence

$$2 \sum_{x} (\nabla I)^T (g_{00}, g_{10})^T ((\nabla I)^T \mathbf{g}(\mathbf{x})\mathbf{d} + \Delta I) = 0, \quad i = 0...n$$

Combining these equations we have

$$\left(\sum_{\mathbf{x}} \mathbf{g}^T (\nabla I)(\nabla I)^T \mathbf{g} \right) \mathbf{d} = - \sum_{\mathbf{x}} \mathbf{g}^T (\Delta I)(\nabla I) \tag{4.7}$$

We may solve this linear equation to find the motion between a pair of images using an iterative refinement process. During each iteration, we warp the initial image according to the current velocity estimate, then we recompute $\Delta I(\mathbf{x})$ using Equation (4.5), and update our incremental estimate of the parameter vector $\mathbf{d}$.

We then accordingly update the optical flow **u**, and recompute the error function (4.6). This process continues until a suitably small error is achieved.

This least squares optimization gives us a simple and fast solution provided that good initial values of the unknown parameters of motion are at hand and a linear representation of motion in terms of some suitable parameters is available.

The key problem remains of how to represent optical flow in terms of motion parameters. Section 4.3 and 4.4 discuss this problem from two different points of view, one from the view of a moving camera imaging a planar surface, and the other from the view of a planar perspective projection of a 3D scene. We show that these are equivalent.

4.3 Planar Surface Rigid Motion

In this section, we review the equations describing the relationship between motion and optical flow in the case of a planar surface undergoing rigid motion. We give both an 8-parameter general model and a more specific 3-parameter model [Bergen92, SzelS97]. When imaging a 3D scene from a fixed viewpoint, rotating the camera imaging a 3D scene is equivalent to imaging a rotating planar surface in 3D space. Thus, this particular case of planar surface motion is relevant to our panorama building problem.

4.3.1 8-parameter General Model

We begin with a general description of a 3D point motion, then to the particular case of planar surface motion. The derivations are referenced from [Horn86], we present them here to compare with another derivation later. Suppose we have a

moving camera in a static environment, with a viewer-centered coordinate frame fixed with respect to the camera, with its Z-axis pointing along the optical axis (see Figure 4.1). The instantaneous velocity of the camera can be expressed in terms of two components, a 3D translation $\mathbf{t} = (t_x, t_y, t_z)^T$ and a 3D rotation $\mathbf{w} = (w_x, w_y, w_z)^T$ about an axis through the origin. Let $\mathbf{P} = (X, Y, Z)^T$ be the 3D position vector of an object point P. Given this motion of the camera, the instantaneous 3D velocity of surface point P is

$$\mathbf{V} = -\mathbf{t} - \mathbf{w} \times \mathbf{P}.$$

We denote the velocity by

$$\mathbf{V} = (\frac{\partial X}{\partial t}, \quad \frac{\partial Y}{\partial t}, \quad \frac{\partial Z}{\partial t})^T.$$

We can rewrite the equation in component form as

$$\dot{X} = -t_x - w_y Z + w_z Y,$$

$$\dot{Y} = -t_y - w_z X + w_x Z,$$

$$\dot{Z} = -t_z - w_x Y + w_y X.$$

The 3D velocities of every surface point in a stationary scene depend on the same rigid body motion parameters given by the above equation.

Each image point p is a perspective projection of a corresponding surface point P, with coordinates given by

$$x = f\frac{X}{Z},$$

$$y = f\frac{Y}{Z}.$$

Here f is the focal length of the projection, i.e., the distance from the camera center to the image center (see Figure 4.1, where it is oc). The derivative of p gives

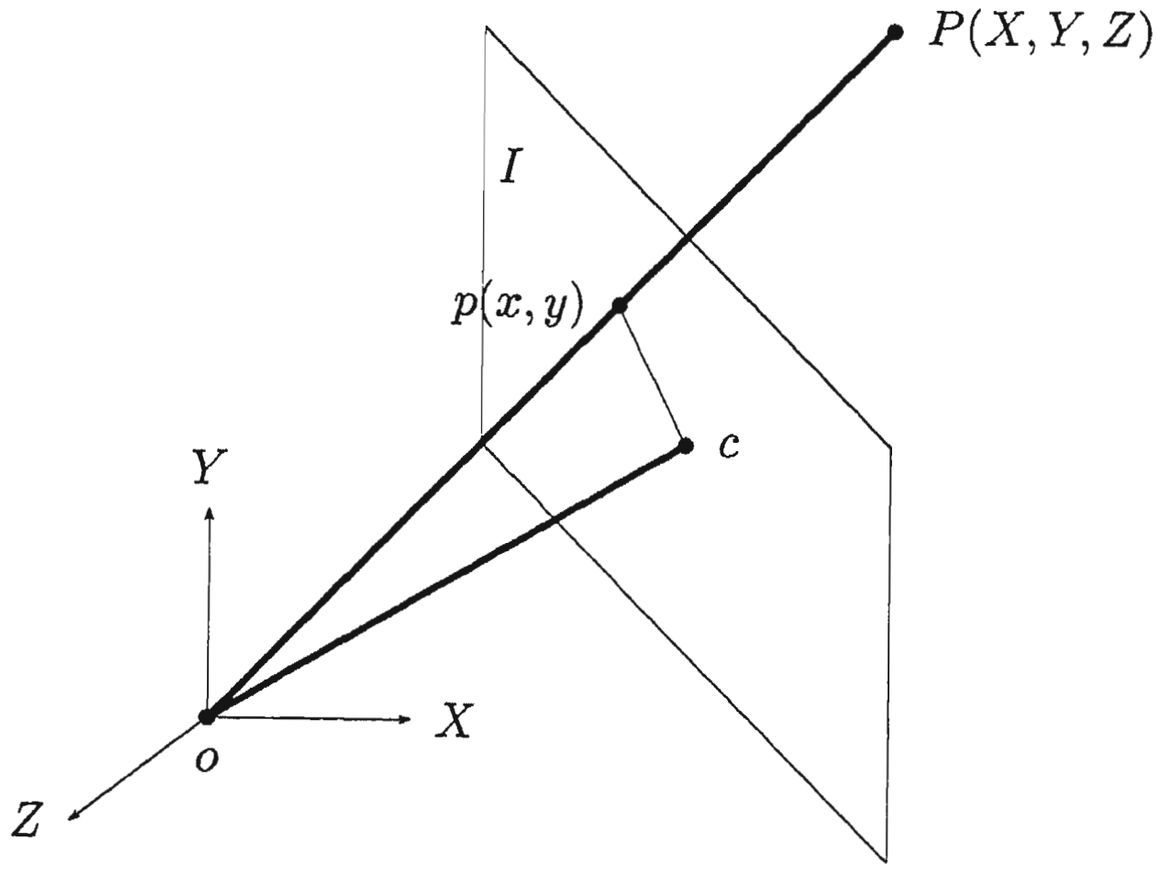


Figure 4.1: Planar Projection

the image velocity or optical flow

$$u = \dot{x} = f(\frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2}),$$

$$v = \dot{y} = f(\frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2}).$$

Substituting for the derivatives of X, Y, and Z yields

$$u = \frac{-ft_x + xt_z}{Z} + w_x(xy/f) - w_y(f + x^2/f) + w_z y,$$

$$v = \frac{-ft_y + yt_z}{Z} + w_x(f + y^2/f) - w_y(xy/f) - w_z x.$$

Rewriting the above equations in matrix form we obtain

$$\mathbf{u}(\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\mathbf{A}(\mathbf{x})\mathbf{t} + \mathbf{B}(\mathbf{x})\mathbf{w} \tag{4.8}$$

where $Z(\mathbf{x})$ is the depth of the image point $\mathbf{x} = (x, y)^T$, and

$$\mathbf{A}(\mathbf{x}) = \begin{pmatrix} -f & 0 & x \\ 0 & -f & y \end{pmatrix},$$

$$\mathbf{B}(\mathbf{x}) = \begin{pmatrix} (xy)/f & -(f+x^2)/f & y \\ (f+y^2)/f & -(xy)/f & -x \end{pmatrix}. \tag{4.9}$$

The matrices $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ depend on the image position and the focal length.

Equation (4.8) shows that optical flow is determined both by the camera motion and 3D object surface geometry. The first term is the translational component of the flow field; it depends on the 3D translation and 3D depth. The second term is the rotational component and depends only on the 3D rotation.

Note here that $\mathbf{u}(\mathbf{x})$ depends at each point on six unknown motion parameters $(t_x, t_y, t_z, w_x, w_y, w_z)$ and an unknown surface $Z(\mathbf{x})$. Thus there are far fewer equations than unknowns. To obtain a solution for camera motion, more assumptions must be made; these are (i) optical flow changes smoothly across the image, and (ii) the object surface is smooth. Under these constraints, a solution to the problem can be found by solving a set of non-linear and over-determined equations [Horn81, Terzopoulos86, Nage87, Tretiak84]. A general solution is hard to obtain, but a solution can more readily be found in some special cases such as pure camera translation or a moving 3D planar surface. A detailed discussion of these issues is beyond the scope of this thesis. However, in particular, we are interested only the planar surface case in the application of stitching panorama mosaics.

The equation for a 3D planar surface is

$$k_1 X + k_2 Y + k_3 Z = 1.$$

Dividing throughout by Z gives

$$\frac{1}{Z} = k_1 \frac{x}{f} + k_2 \frac{y}{f} + k_3.$$

Substituting this into equation (4.8), we have

$$\mathbf{u}(\mathbf{x}) = \mathbf{J_a a} \tag{4.10}$$

where

$$\mathbf{J_a(x)} = \begin{pmatrix} 1 & x & y & 0 & 0 & 0 & x^2 & xy \\ 0 & 0 & 0 & 1 & x & y & xy & y^2 \end{pmatrix}$$

and $\mathbf{a} = (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)^T$ are the combined motion and surface parameters to determine.

Now we can say that, for rigid motion of a planar surface, the optical flow is linearly represented by 8 unknown parameters in a quadratic function of image position of the form in Equation (4.4). A least square solution can be applied directly to the problem by solving Equation (4.7).

Next we consider a less general situation, the pure camera rotation case. We will show that the problem can be reduced to a 3-parameter model with a linear solution.

4.3.2 3-parameter Rotation Model

We now discuss the case assuming that panorama mosaicing is performed from a fixed viewpoint, in which the motion of the camera is undergoing a pure rotation, i.e., panning, tilting and rolling. In this case, the translation vector $\mathbf{t}$ is zero and equation (4.8) simplifies to

$$\mathbf{u(x)} = \mathbf{B(x)w}. \tag{4.11}$$

As noted earlier, the resulting optical flow depends only on the camera rotation speed, and is independent of scene depth. From Equation (4.9), the coefficient matrix $\mathbf{B(x)}$ is determined by image position and focal length. Thus, if the focal length is known, the optical flow components are linearly represented by 3 rotation parameters (w_x, w_y, w_z). Again we may apply a least squares method to the problem by substituting Equations (4.11) and (4.9) into Equation (4.7), which leads to

the linear systems below:

$$\left(\sum_{\mathbf{x}} \mathbf{B}^T(\nabla I)(\nabla I)^T\mathbf{B}\right)\mathbf{w} = -\sum_{\mathbf{x}} \mathbf{B}^T(\Delta I)(\nabla I).$$

As the solution here only involves three rotation parameters, it is a 3-parameter model for aligning images under a pure camera rotation. Because this model has fewer parameters, it converges more quickly to a solution, and is less apt to be trapped in local minimum and thus more stable compared to the 8-parameter model.

In the above we have reviewed the derivation of the 8- and 3- parameter models from the point of a planar surface under rigid motion. Next we present a new derivation that shows the same results can be obtained from a different point of view, i.e. planar perspective projection of 3D scene. This way of derivation can lead to a new solution of 5-parameter model.

4.4 Planar Perspective Projection

We know that imaging rigid motions of a planar surface is equivalent to imaging varying planar perspective projections of a 3D scene, and can be done using an 8-parameter homogeneous transformation [McMillan95]. In constructing a full-view panorama, we assume that the photographs are taken from an approximately fixed viewpoint, so the relation between adjacent images is approximately a planar perspective transformation. The matrix parameters of this transformation are those needed for building up the panorama. In this section, we show that the 8- and 3- parameter models can also be derived assuming the transformation corresponds to a planar perspective transformation, which is a different starting point from previous section. From this perspective view, we further propose a new 5-parameter

model for solving the problem.

4.4.1 8-parameter General Model

Any two images $I'(\mathbf{x}')$ and $I(\mathbf{x})$, which are planar perspective projections of a scene from a common viewpoint, are related by a 2D homogeneous transform $\mathbf{x}' \propto M\mathbf{x}$, i.e.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \propto \begin{pmatrix} mx \\ my \\ m \end{pmatrix} = \begin{pmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix},$$

where $\propto$ indicates proportionality. If we assume that coarse registration has brought I and I' nearly into correspondence, then M is not too far from an identity matrix, and it is convenient to write it in the form

$$M = \mathbf{I} + \mathbf{D} = \begin{pmatrix} 1+d_0 & d_1 & d_2 \\ d_3 & 1+d_4 & d_5 \\ d_6 & d_7 & 1 \end{pmatrix},$$

giving

$$x' = \frac{(1+d_0)x + d_1 y + d_2}{d_6 x + d_7 y + 1},$$

$$y' = \frac{(1+d_3 x) + d_4 y + d_5}{d_6 x + d_7 y + 1}.$$

Thus we may obtain the optical flow as

$$u = x' - x = \frac{d_0 x + d_1 y + d_2 - d_6 x^2 - d_7 xy}{d_6 x + d_7 y + 1},$$

$$v = y' - y = \frac{d_3 x + d_4 y + d_5 - d_6 xy - d_7 y^2}{d_6 x + d_7 y + 1}.$$

Now, d_6 and d_7 in the denominator of the above formula contribute to perspective distortion, and they can be neglected when the transform matrix is nearly an identity matrix. This gives

$$u = d_0x + d_1y + d_2 - d_6x^2 - d_7xy,$$

$$v = d_3x + d_4y + d_5 - d_6xy - d_7y^2,$$

which we may rewrite as

$$\mathbf{u}(\mathbf{x}) = \mathbf{J_d}(\mathbf{x})\mathbf{d} \tag{4.12}$$

where

$$\mathbf{J_d}(\mathbf{x}) = \begin{pmatrix} x & y & 1 & 0 & 0 & 0 & -x^2 & -xy \\ 0 & 0 & 0 & x & y & 1 & -xy & -y^2 \end{pmatrix} \tag{4.13}$$

and

$$\mathbf{d} = (d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7)^T.$$

Substituting these into equation (4.7) yields the linear equation

$$\left(\sum_{\mathbf{x}} \mathbf{J_d}^T(\nabla I)(\nabla I)^T\mathbf{J_d}\right)\mathbf{d} = -\sum_{\mathbf{x}} \mathbf{J_d}^T(\Delta I)(\nabla I).$$

This equation is actually the same formula as derived for the 8-parameter model in Section 4.3.1. Thus, we have derived the same 8-parameter model both by considering a rigid moving plane and a planar perspective transformation.

This 8-parameter model can be further simplified to use less parameters in the transformation matrix when the camera is under going a pure rotation, as we show next.

4.4.2 3-parameter Camera Rotation Model

When the pictures are taken from a fixed viewpoint, the transformation only involves camera panning, tilting, rolling and zooming. If we keep the same focal length, the transformation matrix may be written as (see Section 3.5.1)

$$M = V \cdot R(\theta_y, \theta_x, \theta_z) \cdot V^{-1} \tag{4.14}$$

When M is nearly an identity matrix, we have a small rotation. As $|\theta| \ll 1$, we may approximate $\sin\theta \approx \theta$, $\cos\theta \approx 1$. Thus, Equation(4.14) may be approximated by (see Section 3.5.3)

$$M \approx \begin{pmatrix} 1 & -\theta_z & \theta_y f \\ \theta_z & 1 & -\theta_x f \\ -\theta_y/f & \theta_x/f & 1 \end{pmatrix}. \tag{4.15}$$

From $x' \propto Mx$, we have

$$x' = \frac{x - \theta_z y + \theta_y f}{-\theta_y x/f + \theta_x y/f + 1},$$

$$y' = \frac{\theta_z x + y - \theta_x f}{-\theta_y x/f + \theta_x y/f + 1}.$$

We thus obtain the optical flow equation

$$u = x' - x = \frac{-\theta_z y + \theta_y f + \theta_y x^2/f - \theta_x xy/f}{-\theta_y x/f + \theta_x y/f + 1}$$

$$v = y' - y = \frac{\theta_z x - \theta_x f + \theta_y xy/f - \theta_x y^2/f}{-\theta_y x/f + \theta_x y/f + 1}$$

Again, for a near identity transformation, θ_x and θ_y are small relative to f, so the θ_x/f and θ_y/f in the denominators of the above formulae may be neglected, giving

$$u = -\theta_z y + \theta_y f + \theta_y x^2/f - \theta_x xy/f$$

$$v = \theta_z x - \theta_x f + \theta_y xy/f - \theta_x y^2/f$$

This can be written as

$$\mathbf{u}(\mathbf{x}) = \mathbf{B_d}(\mathbf{x})\theta \tag{4.16}$$

where

$$\mathbf{B_d}(\mathbf{x}) = \begin{pmatrix} -xy/f & (f + x^2/f) & -y \\ -(f + y^2/f) & xy/f & x \end{pmatrix} \tag{4.17}$$

and

$$\theta = (\theta_x, \theta_y, \theta_z)^T.$$

Assuming that the focal length is known, the optical flow is described as a linear combination of three unknown rotation parameters. To apply a least squares method, we substitute the above optical flow into Equation (4.7), giving

$$\left(\sum_{\mathbf{x}} {\mathbf{B_d}}^T (\nabla I)(\nabla I)^T \mathbf{B_d} \right) \theta = - \sum_{\mathbf{x}} {\mathbf{B_d}}^T (\Delta I)(\nabla I).$$

We observe that this 3-parameter solution actually has the same form as that in Section 4.3.2.

We have now given the derivation of both 8- and 3- parameter solutions starting from either imaging a moving planar surface or a varying planar perspective projection of a 3D scene. These all assume a perspective transformation which is only true when the viewing-position is absolutely fixed in the planar perspective projection model, or the surface is absolutely planar in the moving planar surface model. But in real applications of imaging a panorama of a 3D scene, small movements of camera position are hard to prevent. When the viewpoint is not absolutely fixed, theoretically, neither the 8- nor the 3- parameter model is correct, but we can still use them as approximations. When camera translation is small enough in imaging a 3D scene (or equivalently, the surface under motion is near planar), these models are good enough approximations to accommodate

image registration. The quality of the final results depends largely on how close the initial values provided by coarse registration are to the correct solution. We have observed that the 8-parameter model sometimes induces apparent unwanted distortions, while the 3-parameter model induces blurring. These occur due to errors in initial estimate of focal length and some jitter in viewpoint position. In order to remedy such problems, we propose a 5-parameter model next.

4.4.3 5-parameter Camera Rotation Model

We notice in the 3-parameter model that the focal length is assumed to be known. If errors may exist in the value of focal length used, more parameters are needed to allow the focal length to be estimated, too. Let us return to the transformation matrix for camera rotation in Equation (4.15). There are now four unknowns f, θ_x, θ_y, and θ_z. To produce a linear representation, we write the incremental matrix in Equation (4.15) using 5 parameters as follows:

$$M = \begin{pmatrix} 1 & c_0 & c_1 \\ -c_0 & 1 & c_2 \\ c_3 & c_4 & 1 \end{pmatrix}. \tag{4.18}$$

Thus

$$x' = \frac{x + c_0 y + c_1}{c_3 x + c_4 y + 1},$$

$$y' = \frac{-c_0 x + y + c_2}{c_3 x + c_4 y + 1},$$

giving the optical flow equation

$$u = x' - x = \frac{c_0 y + c_1 - c_3 x^2 - c_4 xy}{c_3 x + c_4 y + 1},$$

$$v = y' - y = \frac{-c_0 x + c_2 - c_3 xy - c_4 y^2}{c_3 x + c_4 y + 1}.$$

Because c_3 and c_4 in the denominator are actually θ_x/f and θ_y/f in Equation (4.15), we again omit them assuming the angle of rotation to be small, giving

$$\begin{aligned}\mathbf{u}(\mathbf{x}) &= \begin{pmatrix} y & 1 & 0 & -x^2 & -xy \\ -x & 0 & 1 & -xy & -y^2 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} \\ &= \mathbf{f} \cdot \mathbf{c}.\end{aligned}$$

The corresponding least squares equation is

$$\left(\sum_{\mathbf{x}} \mathbf{f}^T (\nabla I)(\nabla I)^T \mathbf{f} \right) \mathbf{c} = - \sum_{\mathbf{x}} \mathbf{f}^T (\Delta I)(\nabla I).$$

Iterative solution of this 5-parameter model converges faster than the 8-parameter model, but more slowly than the 3-parameter model. It generally provides visually more satisfactory results than either the 3- or 8-parameter solution. Test results and further discussions of each model are given in the next section.

4.5 Discussion and Test Results

In an ideal situation, when there is no translation of the viewpoint and initial estimates of the transformation provided by coarse registration are close enough, all of these models can work well theoretically. The 3-parameter model gives a perfect solution with the fastest speed, and is more robust since it has fewer unknowns; the 8-parameter model generally works well although it converges much more slowly and sometimes gets stuck in local minima, since it contains more free parameters than necessary. The 5-parameter model is somewhere in between in terms of speed and quality.

However, when the viewpoint position is not absolutely fixed and the estimate of focal length is not so accurate, the 3-parameter model often does not register adjacent images perfectly, and some blurring results. While the 8-parameter model can finely register each pixel in the overlap area, often the transformation found is not what we expect, producing unwanted distortions in the non-overlapping region of each image.

In acquisition of the pictures, we assume the user has made a considerable effort to keep a fixed position and only rotate the camera. Therefore, there is mainly a rotation constraint between the pictures thus taken. Moreover, when there are errors in both viewpoint position and coarse registration caused by error in focal length estimation, the 8-parameter model tries to align the images however it can using all its free parameters, and produces unwanted distortions outside the overlap region by deviating too much from the camera rotation constraint. The 5-parameter model overcomes this problem by correcting the focal length while still keeping the rotation constraints, and so provides better results. For the 3-parameter model, there is no way to correct focal length errors, which causes the blurring. We should point out again that all these models are merely approximations when the viewpoint is not absolutely fixed.

Examples are shown in Figures 4.2, 4.3, 4.4 and 4.5. Figure 4.2 gives two original images. Figure 4.3 is the registration result given by the 3-parameter model. We may observe that there is still some parallax causing blurring at the bottom edge of the desk. Results of using the 8-parameter model are shown in Figure 4.4. We can see that the images are finely registered in the overlap region, but outside this region the second image is skewed inwards—see the right-hand edge. Figure 4.5 is the result of using our new 5-parameter model, which is both finely registered and has less skew.

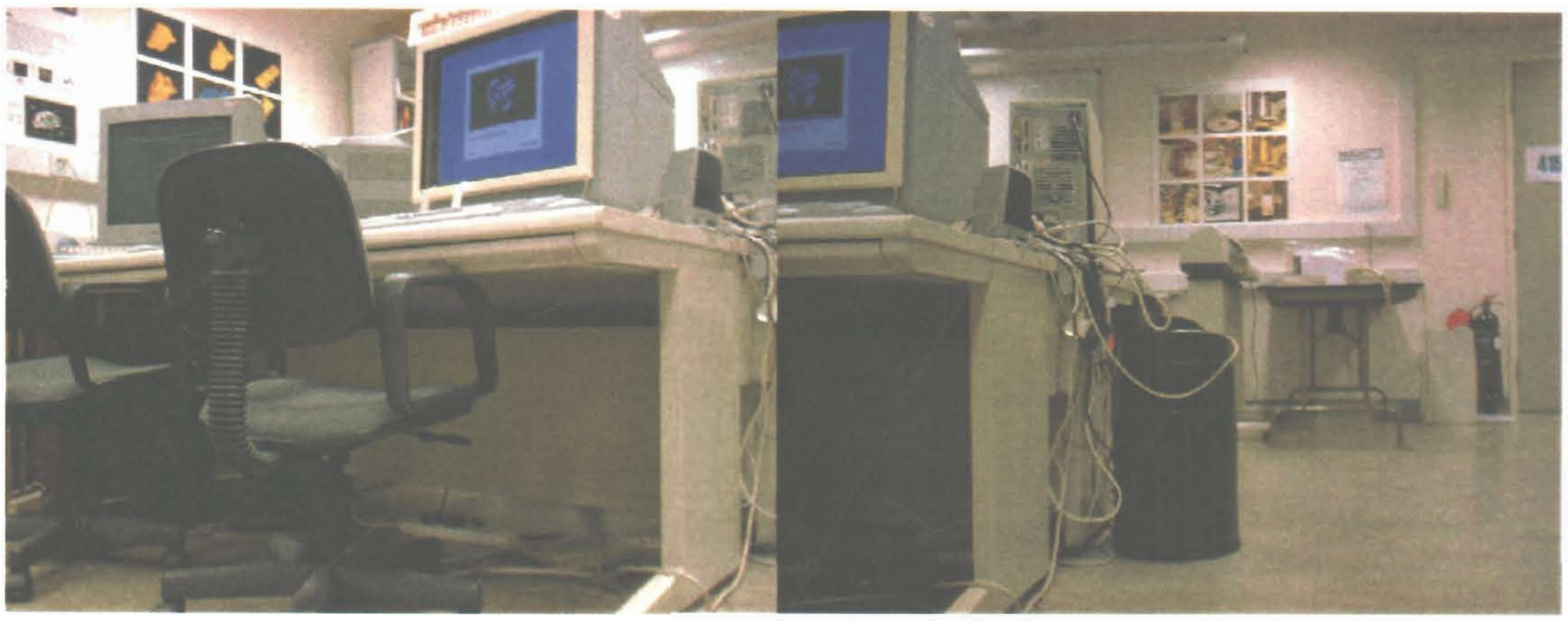


Figure 4.2: Original images for registration

Figure 4.3: Registration using a 3-parameter model. Note blurring at the bottom edge of the desk.

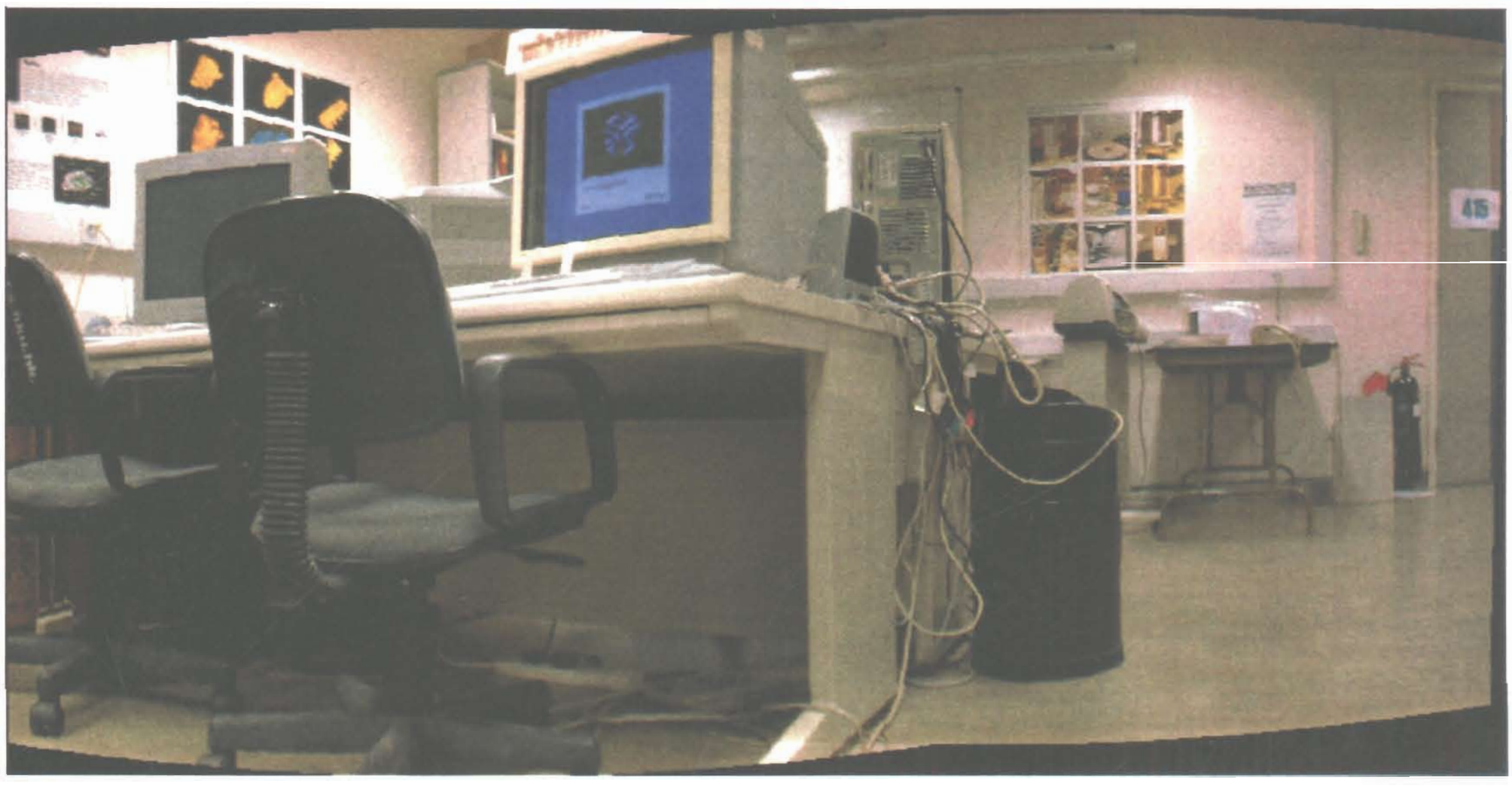


Figure 4.4: Registration using an 8-parameter model. Note skewing at the right edge.

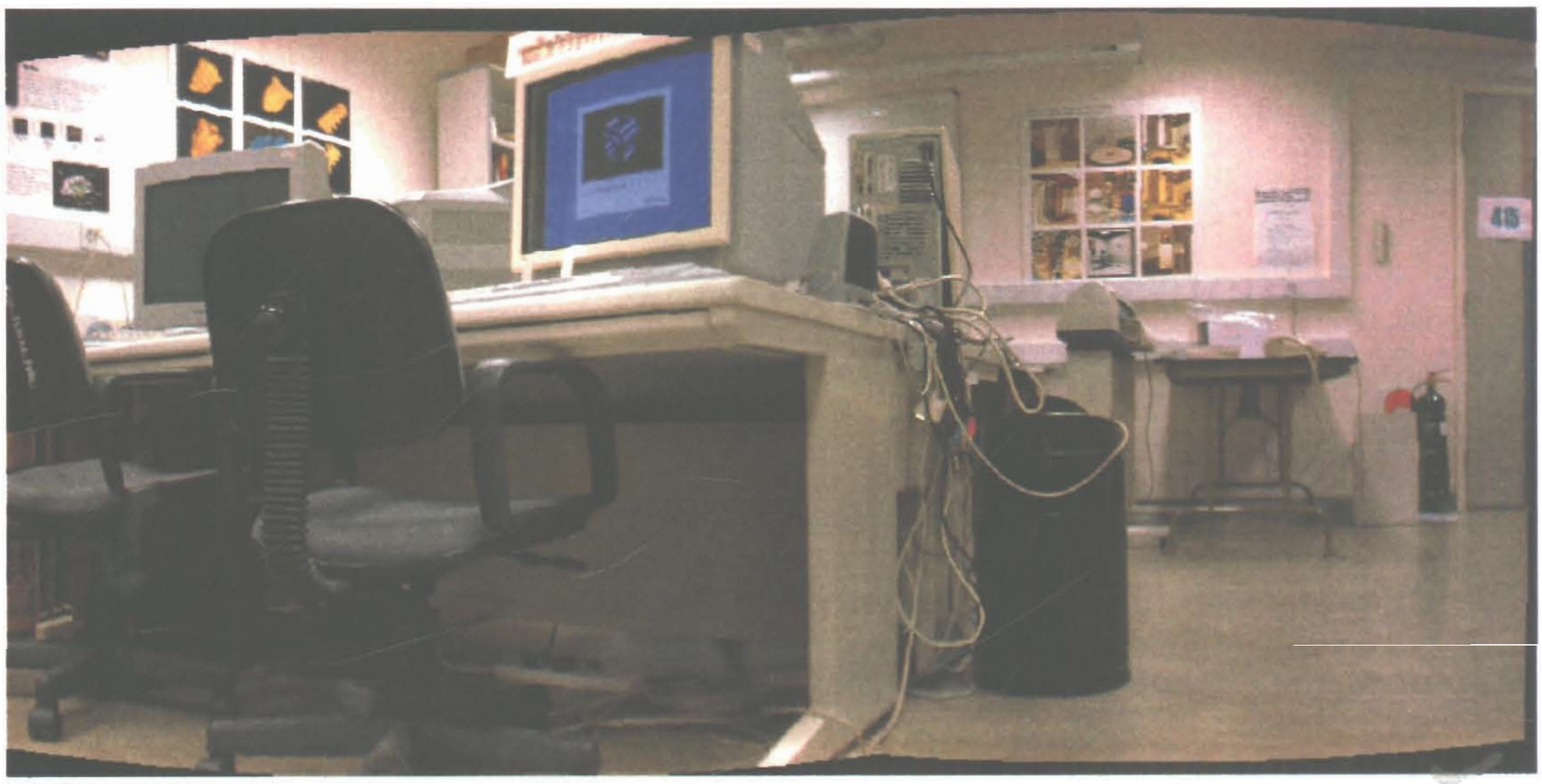


Figure 4.5: Registration using 5-parameter model. There is less blurring and skewing.

There is a problem in the one step linear solution of the 8-parameter model, where we just use four corresponding points to compute a 2D homogeneous transformation matrix directly. The result is sometimes not good even when using synthetic images taken from an absolutely fixed viewpoint. This suggests that errors in the feature location and that these errors are magnified outside the overlap region by the excessive freedom of the model, and hence the method is apt to be ill-conditioned when only a few correspondences are used. An example is shown in Figure 4.6, where the one step solution given by the 8-parameter perspective transformation matrix is illustrated. The four corresponding points are shown connected by red lines. The two pictures are synthetic images with a pure panning of about 30°. We can see that pixels near the four points used to find the correspondence are well registered, but most other pixels further away are not.

Another important issue is over what range of a displacement of pixels between the two images the methods described here still work. The next section gives a discussion of this problem.

4.6 Registration for Large Displacement

The above optical flow based iterative schemes for image registration require that a reasonably good initial alignment must already be known. These methods assume smoothness of objects, uniformity of displacement, and small differences between images. When these conditions are satisfied, they can recover transformations corresponding to a range of a few pixels displacement only. For recovering transformations corresponding to large displacements, a common practice is to perform a hierarchical coarse-fine refinement [Bergen92, SzelS97]. Here we give an alternative way to enlarge the range of displacement for which transformation can be

Figure 4.6: Registration by four tie-points with the 8-parameter model

successfully computed. Note that the methods are used to estimate gradient from a discrete image sequence. Our approach is to apply a Gaussian derivative filter before computing the image gradient. Using a larger deviation factor in the Gaussian filter allows recovery over arbitrarily larger pixel displacements. Although smoothing is often used to improve the performance of optical flow methods, a quantified analysis of the relationship between the smoothing factor and displacement it is possible to recover is not seen in the literature. We perform such an analysis here. We first demonstrate the idea theoretically.

For simplicity, we assume the image displacement is only in the x direction. Applying a first order Taylor series expansion and a least-squares solution to estimate the translational increment δ, we have

$$E(\delta) = \int\int_{\mathbf{x}} (I'(\mathbf{x}+\delta) - I(\mathbf{x}))^2 dxdy \approx \int\int_{\mathbf{x}} (\Delta I + (\nabla_x I)\delta)^2 dxdy \qquad (4.19)$$

where $\Delta I = I'(\mathbf{x}') - I(\mathbf{x})$, and $\mathbf{x}'$ is the displaced image. $\nabla_x I$ is the x derivative of the image. Minimizing the objective function (4.19) with respect to δ leads to the equation

$$\left(\int\int_{\mathbf{x}} (\nabla_x I)^2 dxdy\right)\delta = -\int\int_{\mathbf{x}} (\Delta I)(\nabla_x I) dxdy. \qquad (4.20)$$

Denoting

$$g = -\int\int_{\mathbf{x}} (\Delta I)(\nabla_x I) dxdy, \qquad (4.21)$$

we can write

$$\delta = -\frac{\int\int_{\mathbf{x}} (\Delta I)(\nabla_x I) dxdy}{\int\int_{\mathbf{x}} (\nabla_x I)^2 dxdy} = g/c \qquad (4.22)$$

where $c = \int\int_{\mathbf{x}} (\nabla_x I)^2 dxdy$ is a constant for a given image.

We will show next that g is a function of displacement distance, and can indicate the amount of displacement when it is within $(-4\sigma, 4\sigma)$, where σ is standard Gaussian deviation, assuming Gaussian smoothing has been applied first.

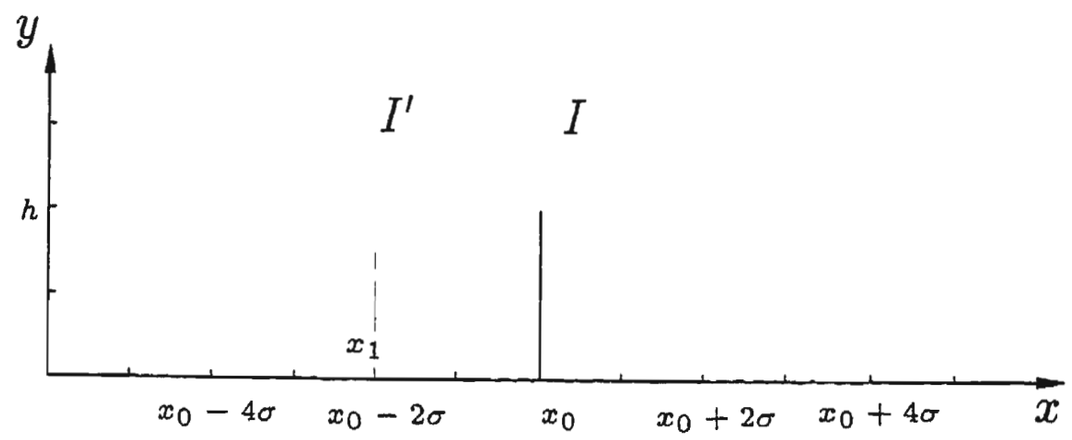


Figure 4.7: A line moving in a picture from x_0 to x_1; h is the line height (number of pixels on the line).

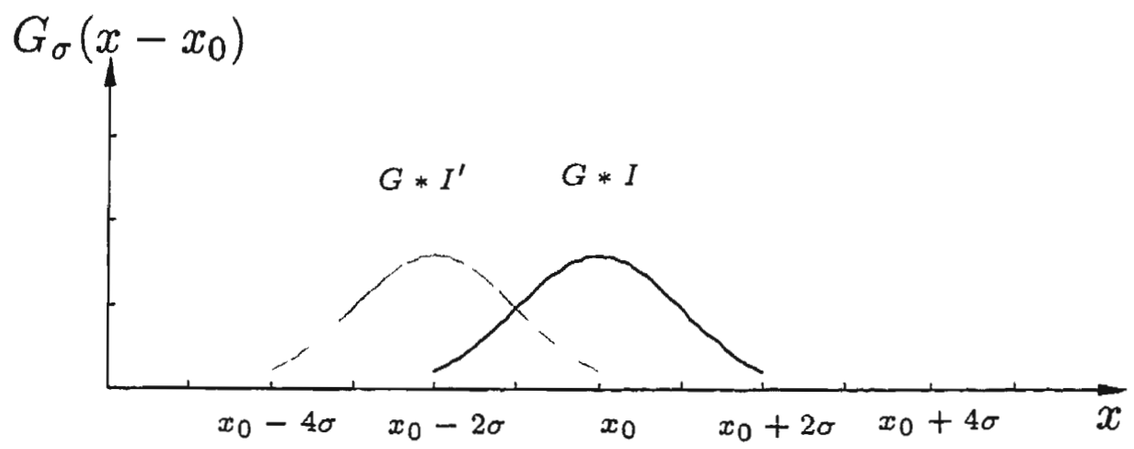


Figure 4.8: Image after Gaussian smoothing; σ is deviation

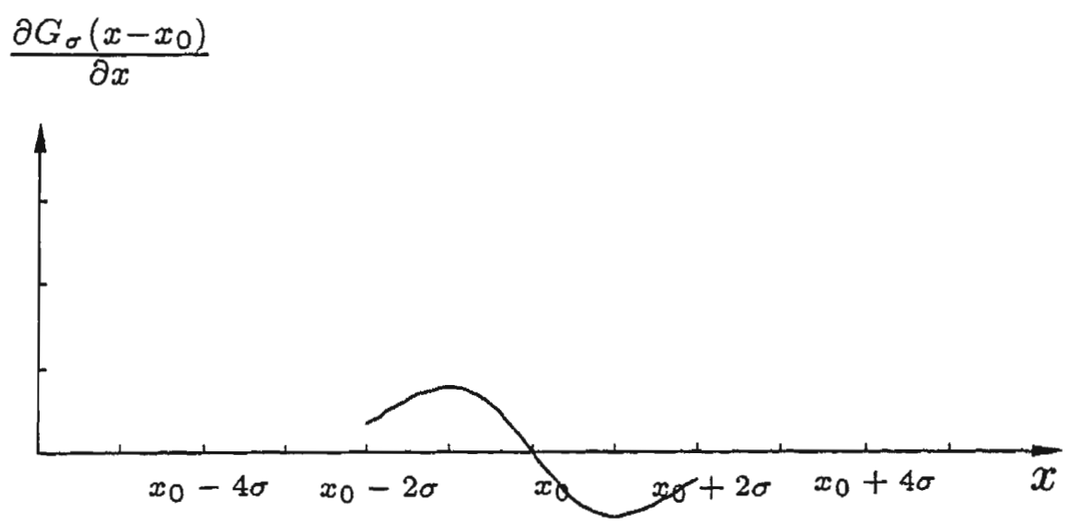


Figure 4.9: x-derivative of Gaussian smoothed image; σ is deviation

Suppose there is only one line in image $I(\mathbf{x})$. The line's motion in the x direction is imaged in $I'(\mathbf{x})$. See Figure 4.7. We assume that

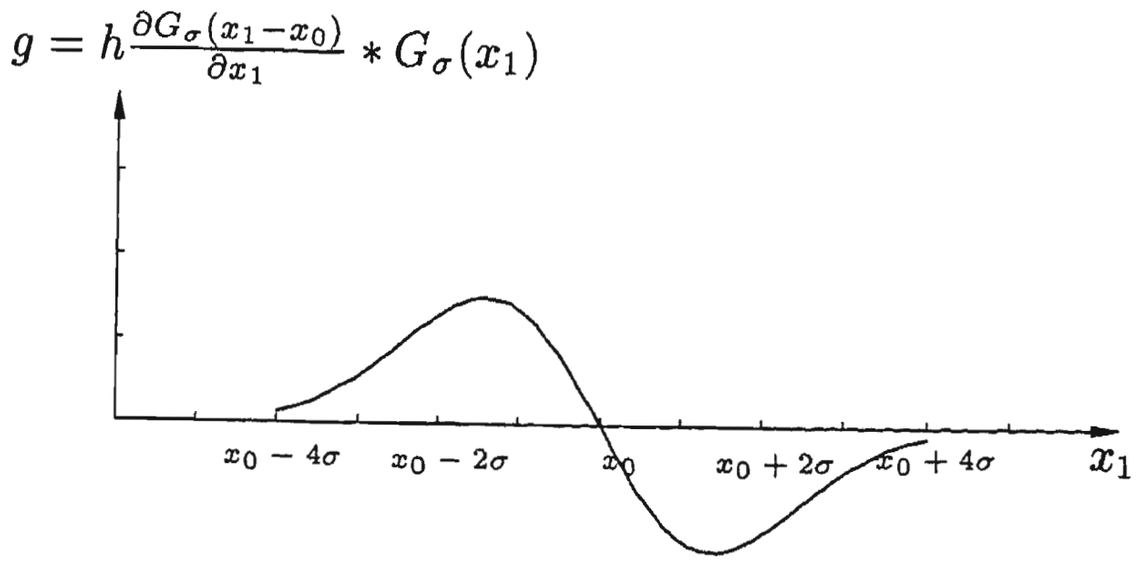


Figure 4.10: Motion indicator g

$$I(\mathbf{x}) = \begin{cases} 1 & \text{if } x = x_0 \\ 0 & \text{otherwise} \end{cases}$$

and

$$I'(\mathbf{x}) = \begin{cases} 1 & \text{if } x = x_1 \\ 0 & \text{if else} \end{cases}$$

First we consider an *unsmoothed* case:

If the two images are unsmoothed as in Figure 4.7, the optical flow formula can not be applied directly. In another words, g in Equation (4.22) does not indicate a meaningful moving direction if applied naively to this case without prior smoothing. As we have used a Taylor expansion to obtain the basic formula for optical flow, this requires the image brightness to be differentiable. If we overlook this condition in discretely sampled images, it may cause a problem.

Since we assume that there is only motion in the x direction only, the gradient

can be calculated by

$$\nabla_x I = I(x,y) - I(x-1,y) = \begin{cases} 1 & \text{if } x = x_0 \\ -1 & \text{if } x = x_0 + 1 \\ 0 & \text{otherwise} \end{cases}$$

whereas

$$\Delta I = I'(x,y) - I(x,y) = \begin{cases} -1 & \text{if } x = x_0, \text{ and } x_1 \neq x_0 \\ 1 & \text{if } x = x_1, \text{ and } x_1 \neq x_0 \\ 0 & \text{otherwise} \end{cases}$$

giving

$$\int\int_{\mathbf{x}} (\nabla_x I \Delta I) dxdy = \begin{cases} 0 & \text{if } x_1 = x_0 \\ -2h & \text{if } x_1 = x_0 + 1 \\ -h & \text{otherwise} \end{cases}$$

Here h is the number of pixels on the line. From Equation (4.21), we have $g \gtrless 0$ for any x_1. That is, the sign of g does not alter when x_1 is either to the right or left of x_0. Hence, no information is given about the direction of motion by g, i.e. the displacement can not be registered, in this case when the image is not smoothed.

Now we examine the *smoothed* case:

Let us smooth the image using a Gaussian filter:

$$G_\sigma(x) = e^{-\frac{x^2}{2\sigma^2}}.$$

Its first order x derivative is

$$\frac{\partial G_\sigma(x)}{\partial x} = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}.$$

The smoothed image $G_\sigma(x) * I(x)$ and its derivative $\frac{\partial G_\sigma(x)}{\partial x} * I(x)$ are shown in

Figures 4.8 and 4.9. We have

$$\begin{aligned}\nabla_x I_\sigma \Delta I_\sigma &= \frac{\partial G_\sigma(x)}{\partial x} * I(x)(G_\sigma(x) * I(x) - G_\sigma(x) * I'(x)) \\ &= \frac{\partial G_\sigma(x-x_0)}{\partial x}(G_\sigma(x-x_0) - G_\sigma(x-x_1)).\end{aligned}$$

Note that $|\frac{\partial G_\sigma(x-x_0)}{\partial x}| = |\frac{x-x_0}{\sigma^2}e^{-\frac{(x-x_0)^2}{2\sigma^2}}|$ has a value small enough for $x < x_0 - 2\sigma$ and $x > x_0 + 2\sigma$, which may be approximated by zero. Thus, Equation (4.21) now becomes

$$\begin{aligned}g &= \int_0^h \int_{x_0-2\sigma}^{x_0+2\sigma} -\nabla_x I_\sigma \Delta I_\sigma dx dy \\ &= -h\int_{x_0-2\sigma}^{x_0+2\sigma} \frac{\partial G_\sigma(x-x_0)}{\partial x} G_\sigma(x-x_0)dx + h\int_{x_0-2\sigma}^{x_0+2\sigma} \frac{\partial G_\sigma(x-x_0)}{\partial x} G_\sigma(x-x_1)dx.\end{aligned}$$

Consider the first term:

$$\begin{aligned}\int_{x_0-2\sigma}^{x_0+2\sigma} \frac{\partial G_\sigma(x-x_0)}{\partial x} G_\sigma(x-x_0)dx &= -\int_{x_0-2\sigma}^{x_0+2\sigma} \frac{x-x_0}{\sigma^2} e^{-\frac{(x-x_0)^2}{2\sigma^2}} e^{-\frac{(x-x_0)^2}{2\sigma^2}} dx \\ &= -\int_{x_0-2\sigma}^{x_0+2\sigma} \frac{x-x_0}{\sigma^2} e^{-\frac{(x-x_0)^2}{\sigma^2}} dx \\ &= \int_{x_0-2\sigma}^{x_0+2\sigma} (e^{-\frac{(x-x_0)^2}{\sigma^2}})' dx \\ &= 0,\end{aligned}$$

and so

$$g = h\int_{x_0-2\sigma}^{x_0+2\sigma} \frac{\partial G_\sigma(x-x_0)}{\partial x} G_\sigma(x-x_1)dx = h\frac{\partial G_\sigma(x_1-x_0)}{\partial x_1} * G_\sigma(x_1) \qquad (4.23)$$

Notice here that x_0 is a constant and x_1 is a variable indicating amount of displacement. The above g is a convolution of $\frac{\partial G_\sigma(x_1-x_0)}{\partial x_1}$ with $G_\sigma(x_1)$, or a Gaussian smoothing of a derivative Gaussian curve. Note that the derivative Gaussian $\frac{\partial G_\sigma(x_1-x_0)}{\partial x_1}$ is effectively non-zero only in the range $x_1 \in (x_0 - 2\sigma, x_0 + 2\sigma)$, which now is extended further left of 2σ and further right of 2σ by the smoothing function $G_\sigma(x_1)$. So g is effectively non-zero for $x_1 \in (x_0 - 4\sigma, x_0 + 4\sigma)$, as can be seen

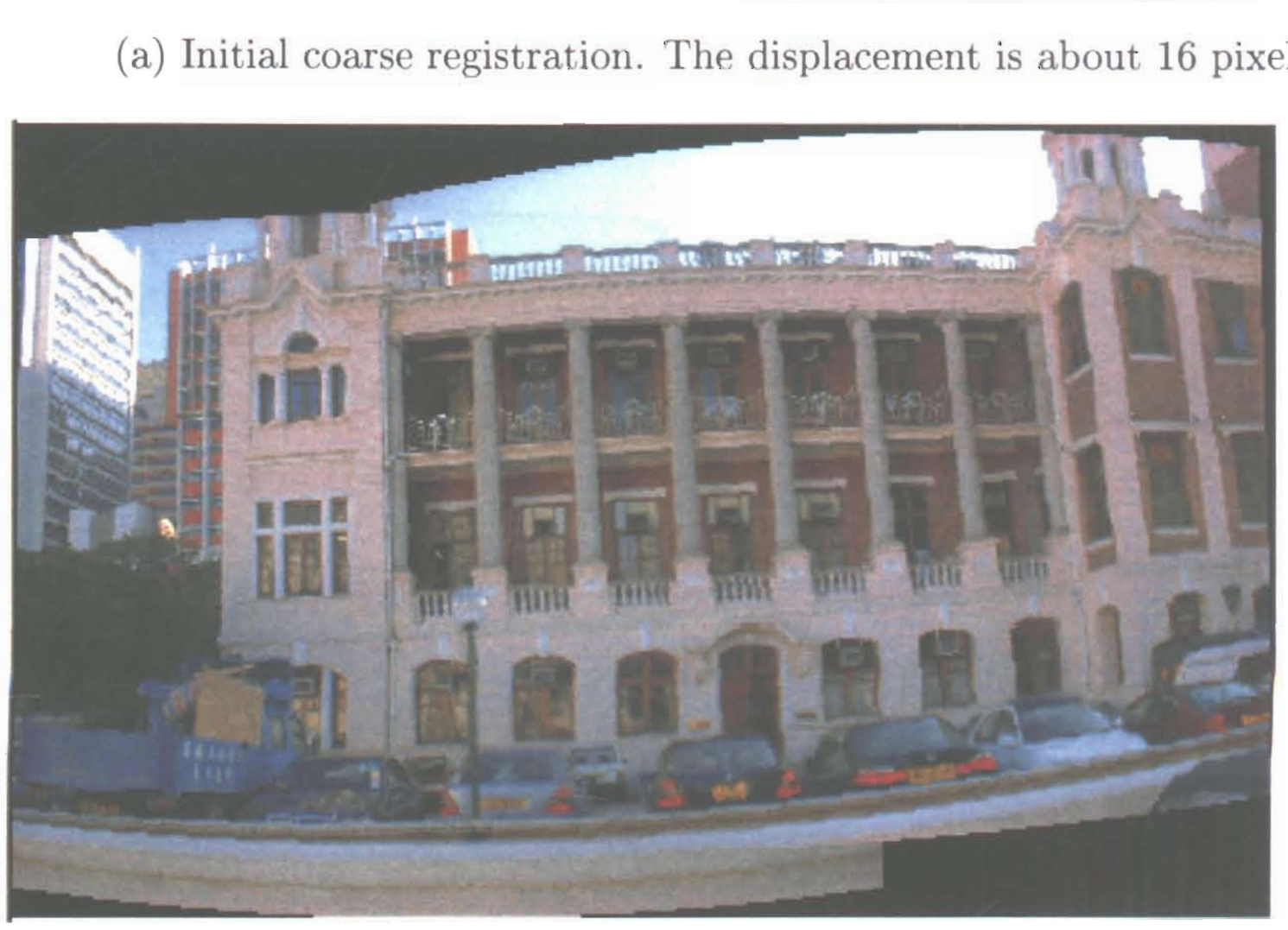
in Figure 4.10. In another word, when the line in $I'(\mathbf{x})$ is moved 4σ away from where it is in $I(\mathbf{x})$, g equals zero and thus from Equation (4.22) that no motion information is available. It shows that theoretically the optical flow based method can recover motion direction within $\pm 4\sigma$ for the simple translational case. Hence, using a large smoothing factor σ can register large displacement.

Thus, from this simple translational example we see that the displacement direction can be recovered within the range of $\delta = \pm 4\sigma$ when we take $g = -\int\int_{\mathbf{x}}(\Delta I_\sigma)(\nabla_x I_\sigma)dxdy$. Similarly, the recovery range is $\delta = \pm 2\sigma$ if only a derivative smoothing filter is applied but no smoothing of the raw images is performed, that is when $g = -\int\int_{\mathbf{x}}(\Delta I)(\nabla_x I_\sigma)dxdy$. Thus, instead of using a coarse-fine hierarchical estimation framework, this suggests a simple but effective approach to enlarging the range of displacements for which optical flow methods can be made to work, i.e., by using a sufficiently large scale derivative filter. This strategy works well for our problem of panorama mosaicking, especially if, as is the case, there are no differences in occlusion between the photographs (because they are taken from a fixed viewpoint). So by simply taking a large filter deviation we may let the major image structures determine the registration. However, the scale of the deviation should not be overly large to result in a poor resolution. We have tested this strategy on a variety of images and obtained good results. As pictures used for panorama building often have relatively small overlap, the speed is reasonably fast.

An illustration of fine registration is shown in Figure 4.11. The displacement is about 16 pixels in (a). Here we take $\sigma = 2.5$. After 12 rounds of iterative refinement, registration converges to the result shown in (b).

Another benefit of using a large deviation filter is that it makes optical flow methods more reliable. Notice that the gradient constraint relies on two approxi-

(a) Initial coarse registration. The displacement is about 16 pixels.



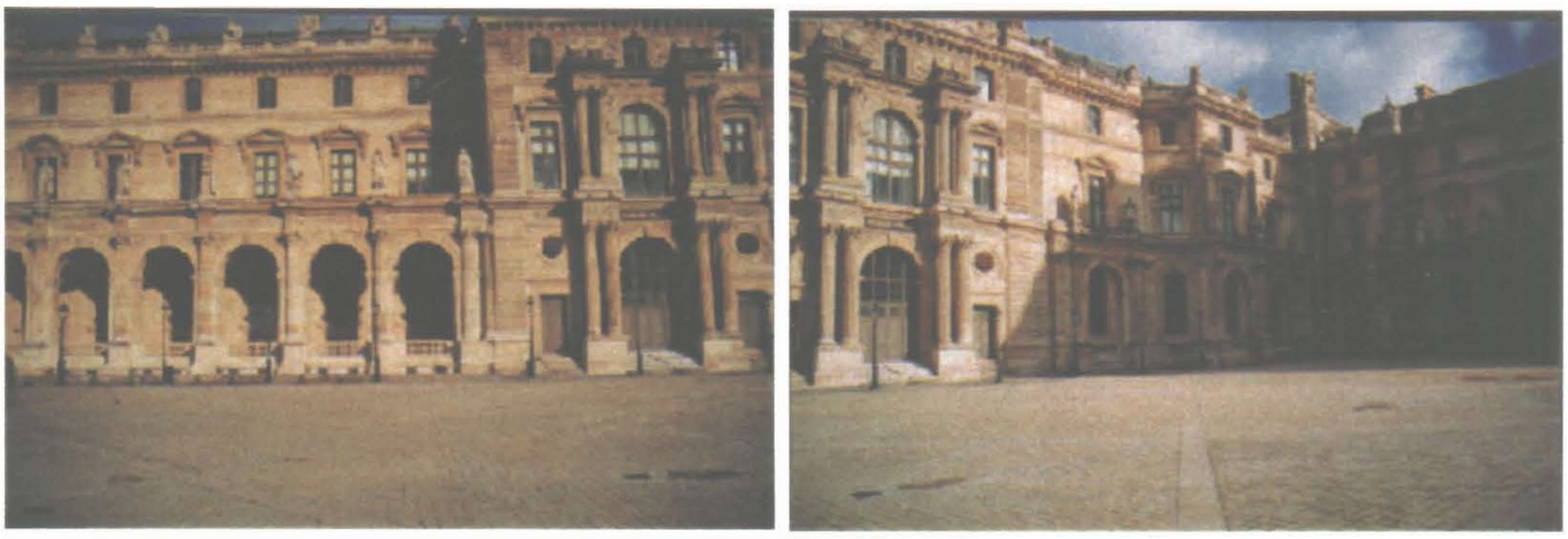
(b) Result of fine registration with $\sigma = 2.5$.

Figure 4.11: Registration of large parallax

mations. One is the image intensity conservation assumption, and the other is the numerical accuracy of the discrete approximation of spatio-temporal first order derivatives. In practice, both assumptions are often only approximately satisfied. However, with appropriate filtering of the raw images one may satisfy the conservation approximation more closely and also improve the accuracy of numerical differentiation. This is particularly important when illumination differences are present. See the example in Figure 4.12. The raw images have considerable lighting differences in the overlap region, see in (a). The displacement is about 12 pixels in (b). We use a deviation $\sigma = 5$. After 16 rounds of iterative refinement, registration converges to the result shown in (c).

4.7 Summary

In theory, only images taken by a camera whose optical center is fixed can be stitched together to produce a panorama with the gradient based registration method described in this Chapter. In practice, when the amount of camera translation is relatively small, the transformations can still be approximated by planar surface motions or perspective transformations. We have shown, starting from these two points of view, the way to linearly express optical flow in terms of model parameters, allowing the latter to be determined using a least squares method. Tests have been conducted on existing 8- and 3-parameter models as well as our new 5-parameter model on images taken with a handheld camera. The proposed 5-parameter model is shown to produce the best quality results. We have also analyzed how the use of a Gaussian filter can be effective in aligning images with large displacement, provided that there are major image structures present in the overlap region and no occlusion present, which is the case for panorama mosaics



(a) The images with apparent lighting difference.

(b) The coarse alignment. The displacement is about 12 pixels.

(c) Result of fine registration with $\sigma = 5.0$.

Figure 4.12: Registration with big lighting differences and large displacement.

taken from a single viewpoint.

The gradient-based registration approach presented in this chapter serves as the fine registration step in our system of building a panorama from images taken with a handheld camera.

Chapter 5

Panorama Building and Tidying

Having registered individual images pairwise, we now need to stitch them together to form a complete cylindrical or spherical panorama. In this chapter, we describe how to build up such a panoramic image by applying the appropriate warping and merging operations to the overlapping adjacent images. We also describe methods for tidying the panoramic image to correct various artifacts resulting from pairwise registration. We focus on the construction and tidying of cylindrical panoramas, and briefly introduce building of spherical panoramas.

5.1 Overview

In building a single viewpoint panorama, a cylindrical model is generally preferred because of its ease of construction, and because it can represent the information about the environment of most interest. The problem of constructing a cylindrical panorama after the adjacent images having been pairwise registered can be stated as follows (the spherical panorama problem is similar).

Given a sequence of images taken from a nearly fixed viewpoint and the relative transformation matrices between them, we want to construct a single cylindrical panoramic image covering a closed horizontal strip of a visible area of a scene. This cylindrical image is used as a texture which is mapped onto a cylinder model centered at the viewpoint, and having a radius equal to the camera focal length. Based on this model, a 3D viewer can be constructed which allows the user to pan around, tilt up and down, and zoom in and out, to view the scene.

The technical difficulties in constructing a cylindrical panoramic image are:

- Due to the presence of tilting rotations, image planes are usually not tangent to the cylinder being mapped onto. This makes the wrapping of images onto the cylinder surface more complicated than in the pure panning case.
- An incorrectly estimated focal length will result in an end gap or overlap when the panoramic image is wrapped onto the cylinder. Measures must be taken to seamlessly close up such an end mismatch.
- The presence of unwanted tilting and rolling in the first image will form a strip on the cylinder in the form of a sine curve, or helix respectively, which should be corrected accordingly.
- Intensity or other differences may exist in irregular shaped overlap regions between adjacent images. These must be smoothed out to achieve a good visual effect.

Previous work on panorama building does not appear to have considered tilting during the warping operation, nor tilting and rolling of the initial image [SzelS97, Shum00, Bao99, Xiong98]. Focal length correction to close the end gap or overlap is simply a stretching or shortening of the composited panorama in one direction [SzelS97, Kang99]. We overcome all these shortcomings in our approach.

To perform a cylinder mapping, firstly, aligned images are projected onto planes tangential to a viewing cylinder; secondly, these projected images are wrapped onto the cylindrical surface.

To smooth out intensity differences between adjacent images where they overlap, we interpolate the intensity of each pixel linearly from the two contributing images according to the pixel's distance from the borders of the overlapping region.

In practice, sampling of images and blending are performed only once at the last stage after the other final corrections mentioned above have been done, as described in Section 5.3.

The rest of this chapter is organized as follows. In Section 5.2, we describe wrapping images onto the cylindrical panorama. Section 5.3 discusses tidying of cylindrical panoramas. Spherical panorama construction is presented in Section 5.4. Section 5.5 summarises the chapter.

5.2 Cylindrical Panoramas

In this section, we formulate the transformations needed for building a cylindrical panorama from the images. To begin with, we show how to determine projection matrices from pairwise registration matrices and the size of the cylindrical map; and specifically, we consider the presence of tilting.

5.2.1 Warping to Tangent Planes

To perform cylinder wrapping, we must first warp the images to planes tangential to the viewing cylinder: see Figure 5.1. Using the automatic registration techniques

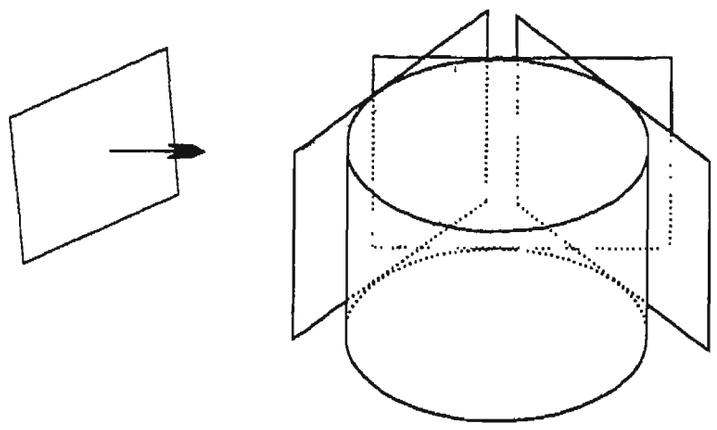


Figure 5.1: Warping an image onto a plane tangential to the viewing cylinder .

in Chapters 3 and 4, we can find a set of matrices [1], $M_0, M_1, ..., M_n$, that relate each pair of adjacent images in a 360° panoramic mosaic sequence. Thus, images in the sequence $I_0, I_1, ...I_n$ are related by

$$I_{i+1} = M_i \cdot I_i, \quad i = 0, 1, ..., n, \quad \text{(indices mod } n\text{)}. \tag{5.1}$$

The cylindrical panorama model is centered at the viewpoint and has a radius equal to the focal length f. We project each image separately to a plane that is tangential to the cylinder. Let the warped images be $I_{W_0}, I_{W_1}, ..., I_{W_n}$. Let the inverse warping matrices be $M_{W_0}, M_{W_1}, ..., M_{W_n}$, defined by

$$I_i = M_{W_i} \cdot I_{W_i}, \quad i = 0, 1, ..., n. \tag{5.2}$$

Note that the inverses, rather than the warping matrices themselves, are used to find each final pixel by sampling.

Let $P_{M_0}, P_{M_1}, ...P_{M_n}$ be the matrices relating two adjacent tangent planes, so

$$I_{W_{i+1}} = P_{M_i} \cdot I_{W_i}, \quad i = 0, 1, ..., n \quad \text{(indices mod } n\text{)}. \tag{5.3}$$

Since there is only a pure panning between adjacent tangent planes, P_{M_i} is determined by a panning angle ϕ_i and the focal length f.

This panning angle ϕ_i is an approximation of the panning rotation of the i^{th} image in the sequence. We estimate this value from the motion of center points of

[1] The subscripts here have a different meaning to those in Equation (3.9).

consecutive images. To do this, ϕ_i, we project the origin of the second image I_{i+1} into the plane of the first image, giving the point (x_{0_i}, y_{0_i}):

$$(x_{0_i}, y_{0_i}, 1)^T = M_i^{-1} \cdot (0, 0, 1)^T.$$

We then project (x_{0_i}, y_{0_i}) into its warped image I_{W_i}, giving the point (x_{w_0}, y_{w_0}):

$$\begin{aligned}(x_{w_0}, y_{w_0}, 1)^T &\propto M_{W_i}^{-1} \cdot (x_{0_i}, y_{0_i}, 1)^T \\ &= M_{W_i}^{-1} \cdot M_i^{-1} \cdot (0, 0, 1)^T, \end{aligned} \tag{5.4}$$

(here $\propto$ means equal up to a scale factor). Finally, the panning angle is given by

$$\phi_i = \arctan(x_{w_0}/f)$$

and the panning matrix relative to the previous image is, from Equation (4.14),

$$P_{M_i} = V_i \cdot R_{P_i} \cdot V_i^{-1},$$

where

$$R_{P_i} = \begin{pmatrix} \cos\phi_i & 0 & \sin\phi_i \\ 0 & 1 & 0 \\ -\sin\phi_i & 0 & \cos\phi_i \end{pmatrix}.$$

Combining Equations (5.1), (5.2) and (5.3) yields

$$\begin{aligned} I_{i+1} &= M_i \cdot I_i \\ &= M_i \cdot M_{W_i} \cdot I_{W_i} \\ &= M_i \cdot M_{W_i} \cdot P_{M_i}^{-1} \cdot I_{W_{i+1}}, \end{aligned}$$

and thus

$$M_{W_{i+1}} = M_i \cdot M_{W_i} \cdot P_{M_i}^{-1}. \tag{5.5}$$

Assuming that the first warping matrix M_{W_0} is an identity matrix which means that the first image has no tilt or roll , (but see later in sections 5.3.1 and 5.3.2), each of the warping matrices can be obtained by applying Equation (5.5).

5.2.2 Wrapping the Images onto the Cylinder

The above matrices are used to calculate the size of the cylindrical image which is obtained by mapping images onto the cylinder and unwrapping it. The cylindrical image is formed by resampling the original images. Denote the output cylindrical image by C. For each pixel $(x_c, y_c) \in C$, we compute the corresponding position (x_w, y_w) in the warped image I_{W_i}, and hence the source pixel (x, y) to be sampled from the i^{th} image I_i, as follows:

$$\begin{aligned} x_w &= f \cdot \tan(\frac{x_c}{f}), \\ y_w &= \frac{y_c}{\cos(x_c/f)}, \\ (x, y, 1)^T &\propto M_{W_i} \cdot (x_w, y_w, 1)^T. \end{aligned} \tag{5.6}$$

5.3 Cylindrical Panorama Tidying

Simply producing a panorama using the above steps results in a cylindrical image with various defects, and these must be corrected to obtain a better final result. Note that, in principle, these problems could be avoided by treating registration of all images simultaneously as a global problem. A global method of alignment for end closing is suggested in [Shum00], but in the objective function used there, the starting and end frames are not explicitly fixed and isolated from the in-between frames so as to impose a closing constraint. In practice, computing a global solution is computationally expensive, and we take a simpler approach to resolving the deficiencies in closing of panoramas constructed using our pairwise registration techniques.

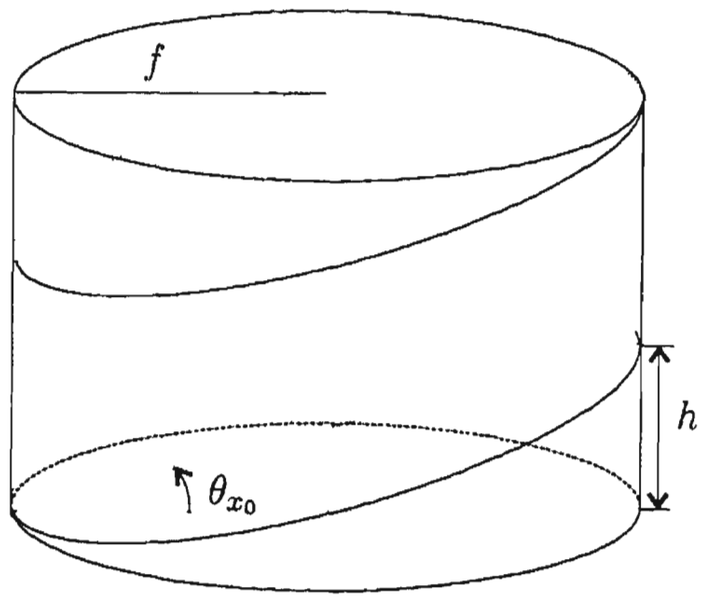


Figure 5.2: Initial tilt correction.

5.3.1 Initial Tilt Correction

At the end of Section 5.2.1, we assumed that the initial warping matrix was an identity matrix. In practice, the first image may be slightly tilted and rolled with respect to the axis of the cylinder. Because of the way the pictures are taken, tilting is likely to be larger than rolling—often there are horizontal reference lines in the scene like the horizon.

In this subsection we consider correction of an initial tilt, and in the next subsection, the effect of an initial roll.

A tilt of the initial image in the sequence will cause the panorama image to change its height on the cylinder as we go around the cylinder, becoming highest half way around (see an example in next chapter Figure 6.8(a)). The result is similar to projection of the images onto an inclined cylinder. The tangent of the initial tilt angle can be estimated by computing the difference in height of the bases of the wrapped images at the start and half way round the cylindrical image, and dividing by the diameter of the cylinder: (see Figure 5.2). Thus

$$\theta_{x_0} = \arctan \frac{h}{2f}.$$

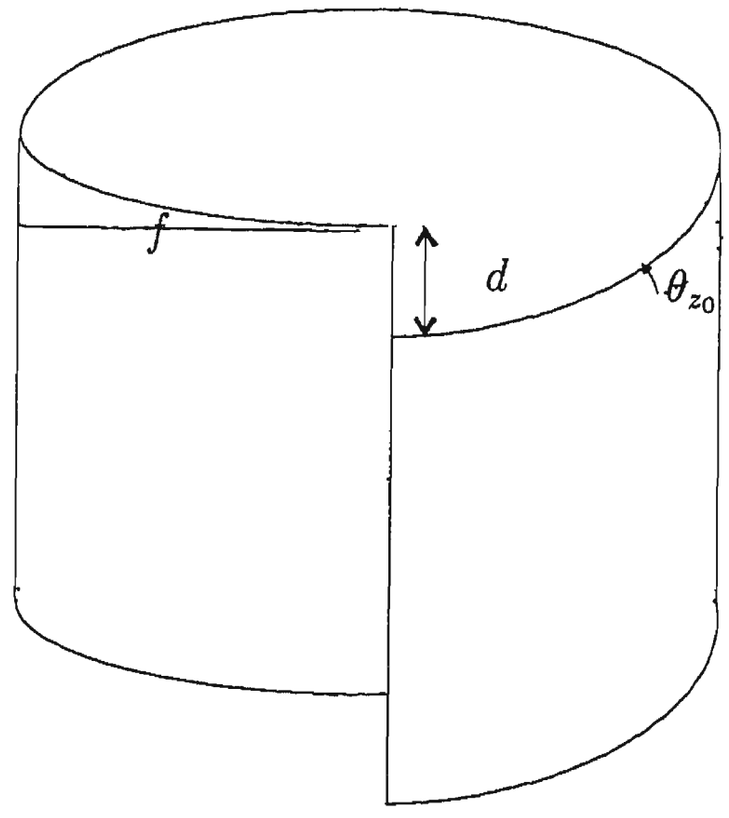


Figure 5.3: Initial roll correction.

5.3.2 Initial Roll Correction

If there is a roll in the orientation of the initial image, then, ignoring any other sources of error, the panorama image will form a helical strip on the cylinder rather than a horizontal strip. By calculating the vertical offset between the initial image and the last image, and dividing by the circumference of the cylinder, we can obtain an estimate for the tangent of the initial roll, (see Figure 5.3), giving

$$\theta_{z_0} = \text{arctg}\frac{d}{2\pi f}.$$

Substituting these two angles into the rotation matrix, a new matrix allowing for the initial tilt and initial roll corrections computed above can be constructed. Replacing the identity matrix M_{W_0} by this new matrix, a more accurate result can be obtained.

5.3.3 Focal Length Correction

Due to errors in focal length estimation and accumulated errors in estimated panning angles between adjacent images, there is usually a mismatch between last and first images when one tries to close up a complete panoramic sequence of aligned images. In the ideal case, the product of the matrices relating adjacent images in a cylindrical image sequence should be an identity matrix, and the total length of the composite image should equal the circumference of the viewing cylinder, with radius equal to the focal length. In other words, when sequentially projecting the images onto the surface of the cylindrical model, the transformation relating the last image in the circular sequence to the first image should leave the trailing edge of the last image exactly adjacent to the leading edge of the first image. Imposing these global closing constraints involves solution of a complicated non-linear problem which does not in general have a closed form of solution. Instead, we propose to solve the problem in a simpler way based on an observation next.

Note that an error in estimated focal length will make the panning angles obtained from the optimization process deviate from the correct value, and result in an end gap or overlap in the composited panorama image. We analyze this effect below.

Focal Length and Panning Angle Error Analysis

For simplicity, we only consider a horizontal line through the center of each image of a pair of adjacent images. See Figure 5.4, where p is the intersection point of the two lines, O is the viewing center, and o_1 and o_2 are the centers of the images, respectively. Let f_0 be the true focal length; suppose we incorrectly overestimate it by an amount Δf. To keep the intersection point unchanged both image planes

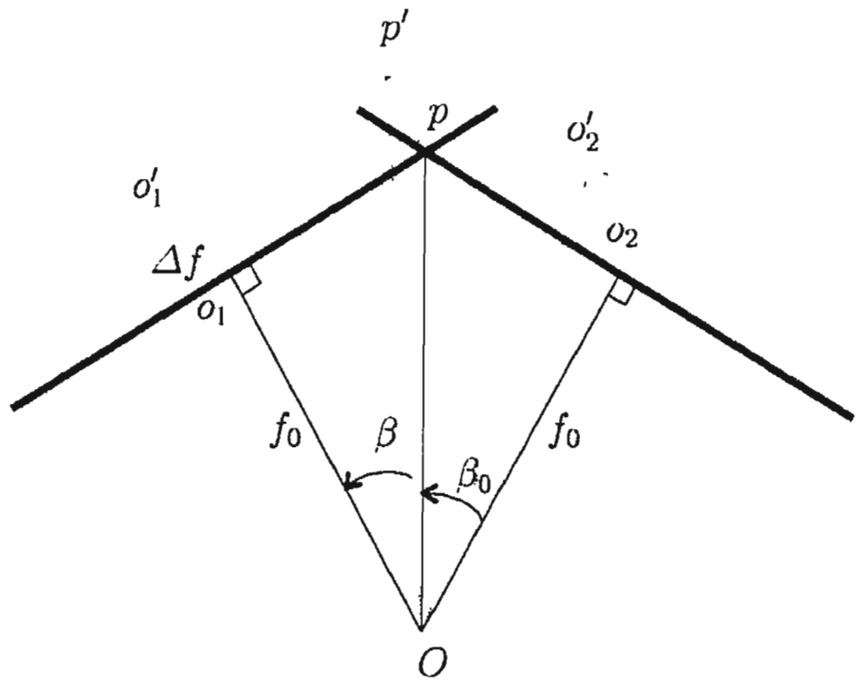


Figure 5.4: Effect of focal length error on panning angle.

should be moved backwards as shown by the dashed lines. Here o_1' and p' are new positions of o_1 and p.

Let $o_1p = c$. Then $o_1'p'$ is also c. Assume $\beta_0 = \angle o_1op$ is half the correct panning angle and β is the value estimated from the incorrectly estimated focal length. Since

$$\text{cotan}\beta = \frac{f_0 + \Delta f}{c}$$

and

$$\text{cotan}\beta_0 = \frac{f_0}{c},$$

we have

$$\begin{aligned} \text{cotan}\beta &= \frac{f_0}{c} + \frac{\Delta f}{c} \\ &= \text{cotan}\beta_0 + \frac{\Delta f}{c}. \end{aligned}$$

Thus

$$\beta = \text{cotan}^{-1}(\text{cotan}\beta_0 + \frac{\Delta f}{c}).$$

The graph of this function is shown in Figure 5.5.

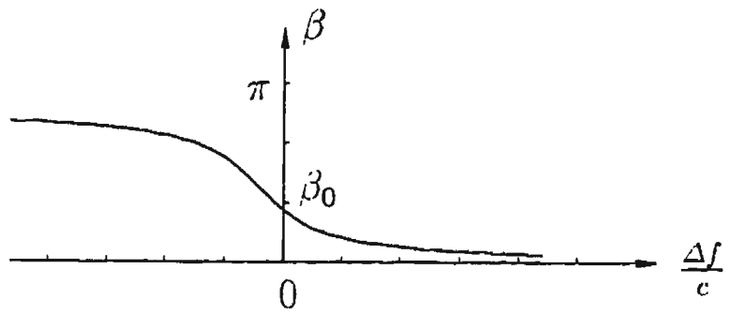


Figure 5.5: Variation of panning angle with respect to focal length error.

Notice that when $\Delta f/c$ is small, using a Taylor series expansion, we can approximately write

$$\beta \approx \beta_0 - \frac{1}{\sin^2 \beta_0} \frac{\Delta f}{c},$$

thus giving

$$\frac{\Delta \beta}{\Delta f} = -\frac{1}{c \sin^2 \beta_0}. \tag{5.7}$$

From Equation (5.7) and Figure 5.5, we see that the error in panning angle varies approximately linearly with focal length error when this error is small. Such errors are accumulated from each image in the circular sequence, and is the main cause of the end mismatch (gap or overlap) in the composited cylindrical image.

We have done experiments to test the effects of focal length error on estimated panning angle produced by our optimization procedure, using four pairs of corresponding points as shown in Figure 5.6. Assuming the panning angle between the images is 60^o, and that each image has the same viewing angle of 90^o, we let the corresponding points be the projections of the object points A, B, C, and D. Then, $a = (f \tan 15°, 0) \Leftrightarrow a' = (-f, 0)$, $b = (f, 0) \Leftrightarrow b' = (-f \tan 15°, 0)$, $c = (f \tan 30°, 100) \Leftrightarrow c' = (-f \tan 30°, 100)$ and $d = (f \tan 30°, -100) \Leftrightarrow d' = (-f \tan 30°, -100)$ (image points for C and D are not shown). We let Δf vary between $(-50, 50)$ given focal lengths of 150, 200, 250 and 300 respectively. Using

the optimization Equation (3.14), we obtain four graphs of the estimated panning angle as shown in Figure 5.7. (When $|\Delta f|$ exceeds 50, an unrealistically large error for f in the above range, the optimization procedure does not always converge.)

Figure 5.7 shows that the panning angle error is nearly linear with respect to changes in focal length, as predicted, even when the error in focal length is quite large relative to the focal length itself. The optimization procedure compensates for the error in focal length by producing an error in panning angle. Successive errors between each pair of adjacent frames are accumulated, resulting in a large error of composition length while pairwise registration still looks good.

The above error analysis is used to determine how to carry out focal length correction as described next.

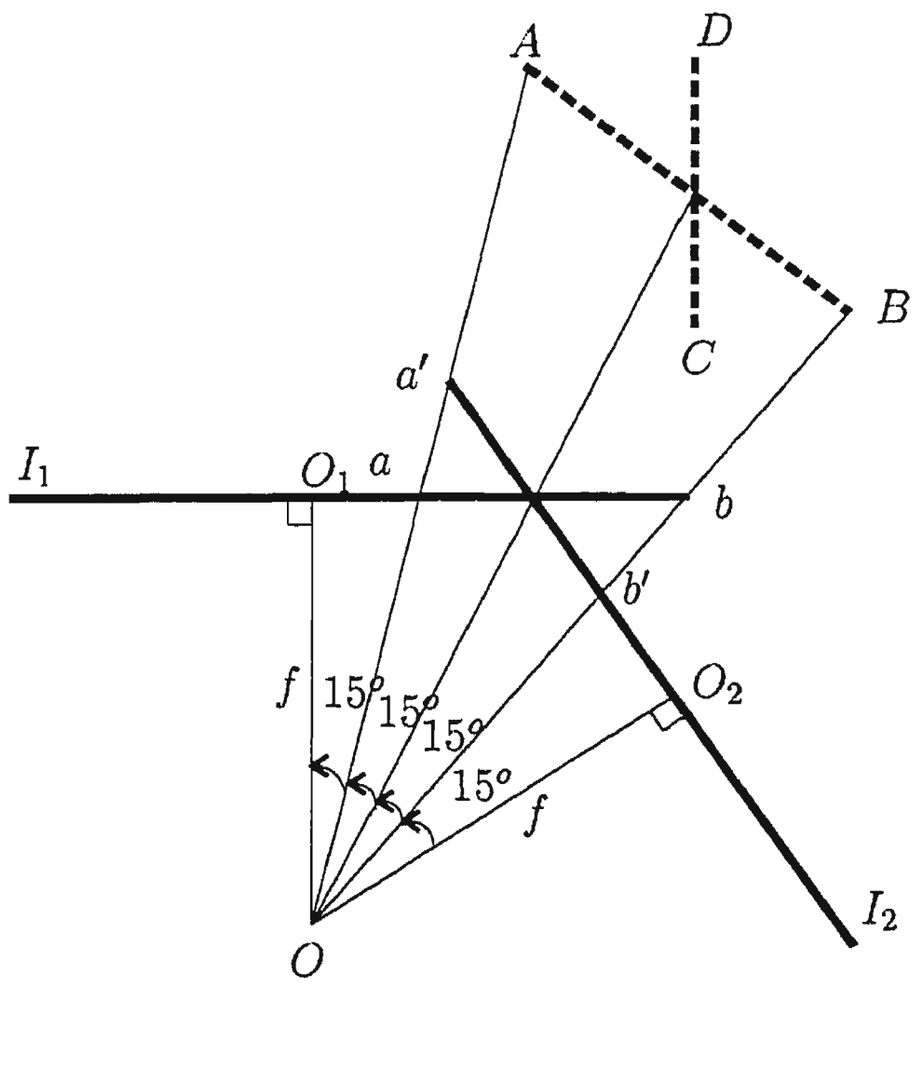


Figure 5.6: Image points corresponding to object points A, B, C and D.

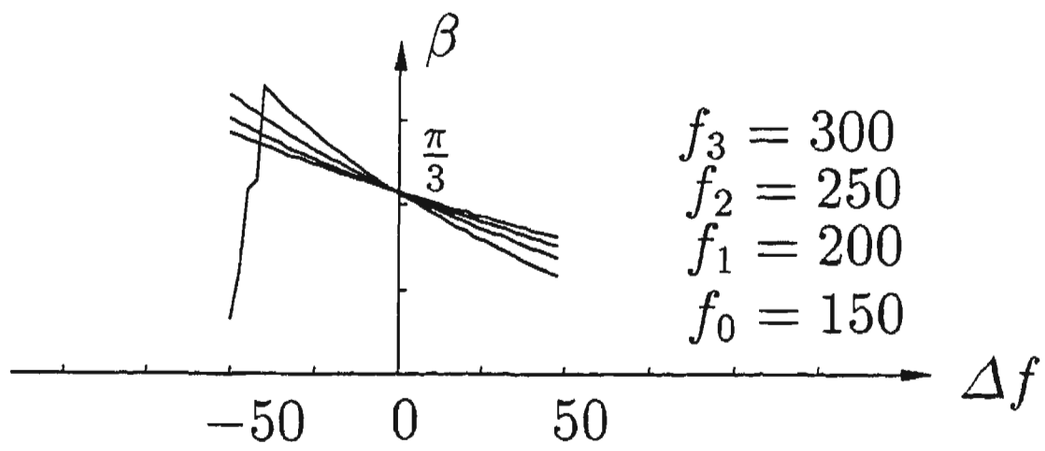


Figure 5.7: Panning angles resulting from optimization in the presence of focal length errors.

Correction—Gap Closing

As shown above, our registration algorithm can give a result that still looks fine even when a considerable error of focal length is present, while this error may result in a relatively large error in end matching. In other words, the overall length of the composite image is much more sensitive to focal length error than pairwise registration errors between adjacent images. Thus, we use the determined composition length to refine the focal length, while keeping the local registration.

Let L be the length of the composite panorama. The viewing cylinder has a radius $L/(2\pi)$, which should be equal to the focal length. The focal length obtained from $f' = L/(2\pi)$ is in general different from the value f estimated in Chapter 3. Thus, we replace the focal length f by f'.

In turn, the panning angles between pairs of adjacent images need to be updated accordingly, so that the overlap region between any two aligned images remains unchanged. We iteratively re-estimate the focal length and consequently update the panning angles several times until convergence is obtained. This procedure closes any horizontal gap or overlap between the last and first images caused by mis-estimation of focal length. Details of the process are as follows.

See Figure 5.8 showing a pair of adjacent images. Let O be the viewpoint,

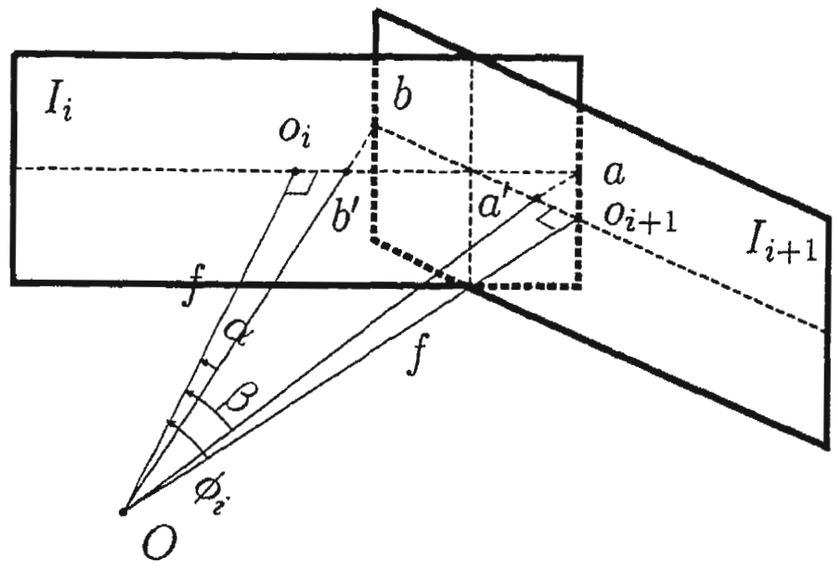


Figure 5.8: Focal length correction.

and o_i and o_{i+1} be the origins of images I_i and I_{i+1}, located at the centre of each image. We assume that each image has width w and the same true focal length f, i.e. distances $Oo_i = Oo_{i+1} = f$ and $o_i a = o_{i+1} b = w/2$, where a is the mid-point of image I_i's right edge, b is the mid-point of image I_{i+1}'s left edge, and w is the width of each image. Let the overlap length be l_i, i.e. $b'a = ba' = l_i$, where b' is the projection of b in image I_i, and a' is the projection of a in image I_{i+1}. From Figure 5.8, we have

$$\tan\alpha = \frac{w/2 - l_i}{f},$$

$$\tan\beta = \frac{w}{2f},$$

$$\phi_i = \alpha + \beta,$$

giving

$$l_i = \frac{w}{2} - f \cdot \tan(\phi_i - \arctan(\frac{w}{2f})). \tag{5.8}$$

This overlap length is determined by the registration process, and should not be altered when the panning angles are updated.

Let θ_g be the angular error in closing the panorama given by:

$$\theta_g = 2\pi - \sum_{i=0}^{n} \phi_i \tag{5.9}$$

where a positive angle denotes a gap in the final panorama and a negative angle, an overlap.

We now update the estimate of focal length using

$$f' = \frac{2\pi - \theta_g}{2\pi} \cdot f, \tag{5.10}$$

and then from Equation 5.8, update each panning angle to be

$$\phi_i' = \arctan(\frac{w/2 - l_i}{f'}) + \arctan(\frac{w}{2f'}).$$

We then go back to Equation (5.9) and repeat this process until $|\theta_g|$ reaches a small enough value.

This process gives an increment for each panning angle $\Delta\phi_i$ which is the difference between the final and original panning angle for each pair of adjacent images. Using these and the new focal length f', together with the previously determined rotation matrices R_i using registration techniques in Chapters 3 and 4, we can compute revised values for each alignment matrix M_i using

$$M_i' = \begin{pmatrix} f' & 0 & 0 \\ 0 & f' & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos\Delta\phi_i & 0 & \sin\Delta\phi_i \\ 0 & 1 & 0 \\ -\sin\Delta\phi_i & 0 & \cos\Delta\phi_i \end{pmatrix} \cdot R_i \cdot \begin{pmatrix} 1/f' & 0 & 0 \\ 0 & 1/'f & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

On performing the stitching procedure with these new matrices, the cylindrical panorama will be closed.

Note that the simpler approach of just using Equation (5.10) to force closure of the cylinder surface as used in [Kang99, SzelS97] is not accurate when the pairwise registration is obtained from the 3-parameter rotation model, since incorrect estimates of focal length induces errors in panning angles which are not corrected by the Equation. When the 5- or 8- parameter model is adopted in pairwise registration, using Equation (5.10) is an adequate approximation.

5.3.4 Deskewing

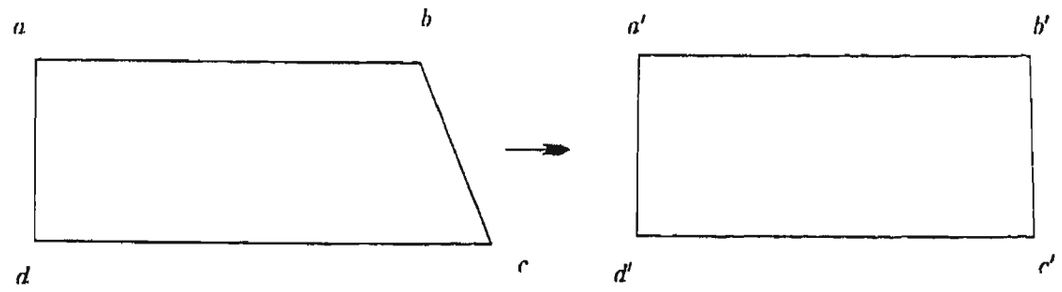


Figure 5.9: Minor error in unwrapped panorama.

Even after all these corrections have been made to close the panorama, there may still be some small mismatches at the end due to other reasons such as imprecise model assumptions, e.g. the camera position moves, or accumulated errors in matrix calculations In particular, the unrolled panorama may form a quadrilateral rather than a rectangle: see Figure 5.9. We correct such a problem using a deskewing method. Note that any quadrilateral can be warped into a rectangle by a projective transformation. Therefore we can warp the four corners of the unrolled panorama to impose closure. Finding the necessary transformation is a straightforward calculation given the initial and desired final corners of the panorama.

Although this treatment lacks a sound theoretical basis, the distortion it induces over the whole image is much less apparent to the viewer than a small gap or overlap at the ends of the panorama would be. By doing so, the remaining end mismatches are distributed over the whole panorama thus reducing their visible effects.

5.3.5 Blending

In practice, the final panorama may have discontinuities in intensity (shadow or ghost effects) if we simply use an unweighted averaging to combine adjacent images in their overlap region. This is caused by various reasons such as incompatible

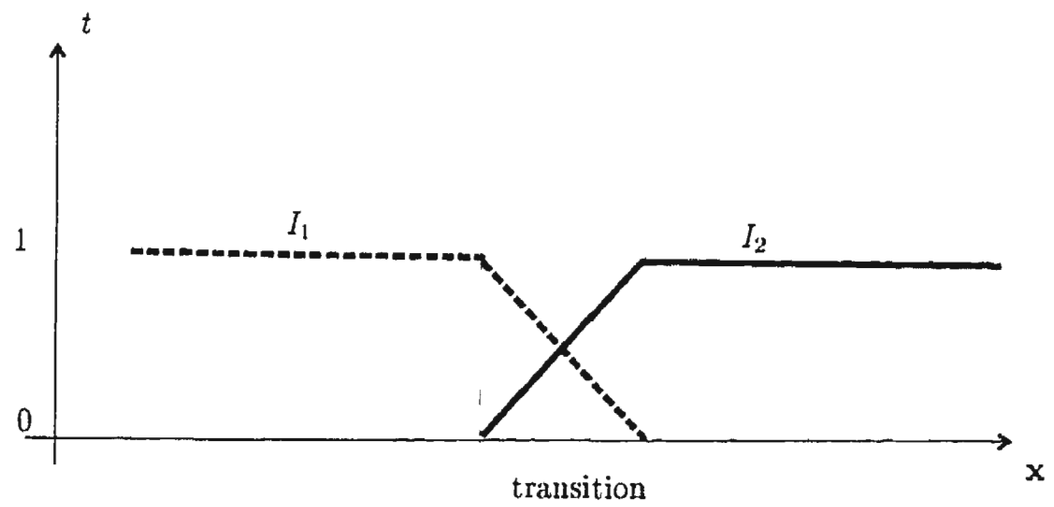


Figure 5.10: Blending of overlap.

model assumptions in pairwise registration or uneven exposure of the adjacent photographs or changing lighting conditions. Human eyes are very sensitive to these artifacts.

To reduce such discontinuity effects in the final panorama, we apply a simple blending algorithm. Pixel values in each region where two photographs overlap are computed by linearly interpolating the intensity values of corresponding pixels from contributing images weighted according to their distance from the borders of the overlap region. Thus, we use

$$I_c(x,y) = t(x,y) \cdot I_1(f_1(x), g_1(y)) + (1 - t(x,y)) \cdot I_2(f_2(x), g_2(y))$$

where (x, y) is a pixel in the overlap region; f_1, g_1, f_2, and g_2 are the warping functions of the two images I_1 and I_2 onto the cylindrical image I_c, respectively, see Equation5.6; $t(x, y)$ is the interpolation coefficient ranging between 0 and 1. For simplicity we assume here the transition length is 1. See Figure 5.10.

Using the transformation matrices obtained from pairwise registration in the last two chapters, corrected as in the previous sections of this chapter, we can compute the size of an empty cylindrical panorama. To begin with, we wrap the first image onto this empty panorama without any blending since there is no content in it. Then we wrap and blend subsequent images onto it one by

one. In our application we have assumed a small overlap between adjacent images, which will result in only two images meeting in any overlap region. In practice, our method can handle more than two overlapping images, it has a drawback of uneven contributions from each image. In summary, we find that this simple blending method works well in practice and produces good visual effects. More complicated blending approaches can be found in [Xiong98, Burt83], where a labeling weighted scheme or a multi-resolution spline blending algorithm is used.

5.4 Spherical Panoramas

To represent an enclosed environment, a spherical panorama can be used. With an ordinary camera, several sequences of images need to be taken to cover the whole space. Normally, at least three sequences must be taken: upper, middle and lower sequences. The middle strip is panning horizontally, while the upper and lower strips are panning with a tilting angle up or down. The top and bottom are covered by the overlaps in the upper and lower strips. These sequences must be registered and stitched in both horizontal and vertical directions to make a panorama. A multi-image registration problem is involved in the spherical case which is beyond the scope of this research. With a known transformation between the images, we study the stitching problem. In the rest of this section, the appropriate spherical panorama warping operations are described.

Suppose three sequences of images have been taken in order to construct a spherical panorama. Assume the transformations between the adjacent images in the middle strip are known, and the transformations between images in the upper (or lower) and the adjacent images in the middle strip are given. For the middle sequence of images, the stitching is performed horizontally in a cyclical sequence; for the upper and lower sequences of images, stitching is performed vertically with

their adjacent images in the middle sequence. The problems to be solved are basically the same as before: to find the warping matrices between the images and planes tangent to the spherical surface given the results of pairwise registration, and to project each source image onto the spherical model via its tangent plane.

5.4.1 Warping to Tangent Planes

To perform spherical wrapping, the images must first be warped onto planes tangential to the viewing sphere. The formulation of the warping is the same as for cylindrical warping except for the matrix P_{M_i}, which relates two adjacent tangent planes. Note that in the spherical case, not only a panning angle but also a tilting angle is involved.

To find the panning and tilting angles, we project the origin of the second image I_{i+1} into the plane of the first image. In a similar way to cylindrical warping, the panning and tilting angles are given by

$$\phi_{y_i} = \arctan(x_{w_0}/f),$$

$$\phi_{x_i} = \arctan(y_{w_0}/f),$$

where (x_{w_0}, y_{w_0}) is given by Equation 5.4.

The panning matrix relative to the previous image thus is

$$P_{M_i} = V_i \cdot R_{P_i} \cdot V_i^{-1},$$

where

$$R_{P_i} = \begin{pmatrix} \cos\phi_{y_i} & 0 & \sin\phi_{y_i} \\ 0 & 1 & 0 \\ -\sin\phi_{y_i} & 0 & \cos\phi_{y_i} \end{pmatrix} \cdot \begin{pmatrix} \cos\phi_{x_i} & 0 & -\sin\phi_{x_i} \\ 0 & 1 & 0 \\ \sin\phi_{x_i} & 0 & \cos\phi_{x_i} \end{pmatrix} \cdot$$

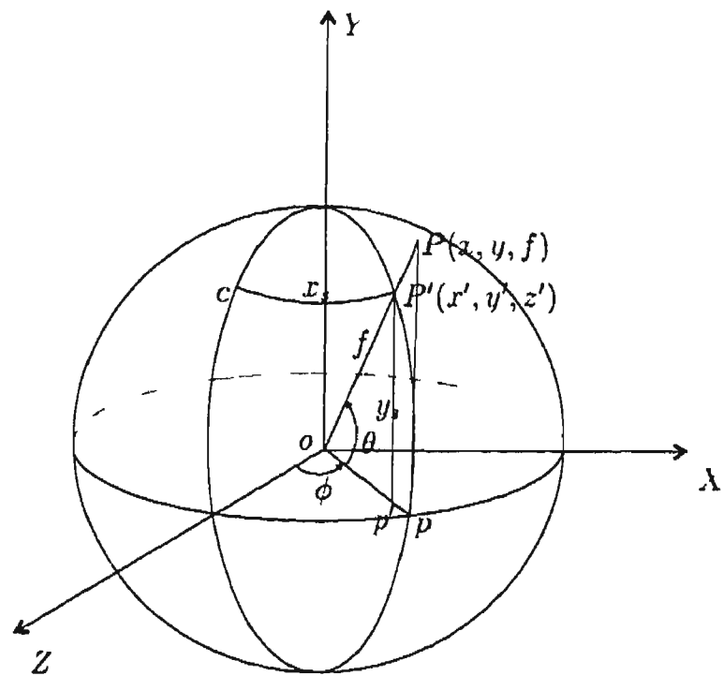


Figure 5.11: Spherical warping.

5.4.2 Warping Images onto Sphere

Now we unwrap the spherical surface to a 2D panorama image plane, denoted as I_s, for compact storage of the panorama. The unwrapping is done as follows: using horizontal planes to cut the sphere from top to bottom, the spherical surface is thus sampled by a set of parallel circles; we then flatten this set of circles onto a 2D image plane, in which the top and bottom middle pixels correspond to the top and bottom points of the surface. For simplicity, we assume that all images are tangent to the spherical model. Let the center of the first image be the origin of coordinates of the image plane I_s, and the unwrapped equator of the spherical surface be its x-axis. We demonstrate how to map a point of an image onto this unwrapped 2D plane representing the spherical surface.

Let $p(x, y)$ be a point of image I_i and f be the radius of the sphere. The point's original 3D coordinates can be written as (x, y, f): see Figure 5.11. After panning and tilting rotations, $R_i = P_{M_i} P_{M_{i-1}} ... P_{M_0}$, the point is transformed to

$$(x', y', z')^T = R_i (x, y, f)^T. \tag{5.11}$$

Since only the center of each image lies on the spherical surface, we need to project the point (x', y', z') onto the spherical surface using

$$T_1 : \begin{cases} \bar{x} = x'f/\sqrt{(x'x' + y'y' + z'z')} \\ \bar{y} = y'f/\sqrt{(x'x' + y'y' + z'z')} \\ \bar{z} = z'f/\sqrt{(x'x' + y'y' + z'z')}, \end{cases} \tag{5.12}$$

where $(\bar{x}, \bar{y}, \bar{z})$ is the point's projection on the spherical surface. Representing the point in spherical coordinates (θ, ϕ) yields

$$T_2 : \begin{cases} \theta = \arcsin(\bar{y}/f) \\ \phi = \arctan(\bar{z}/f). \end{cases} \tag{5.13}$$

Let the corresponding point in the unwrapped panorama image be (x_s, y_s). From Figure 5.11, we have

$$T_3 : \begin{cases} x_s = f\cos\theta(\phi - \pi/2) + f\pi/2 \\ y_s = f\sin\theta. \end{cases} \tag{5.14}$$

Combining the above Equations (5.11), (5.12), (5.13) and (5.14), we obtain the transformation that maps a point in the image onto the unwrapped spherical panorama

$$T : (x, y) = T_3 T_2 T_1 R_i (x, y, f)^T. \tag{5.15}$$

5.5 Summary

We have presented in this chapter the warping operations for constructing cylindrical and spherical panoramas. We have analyzed the effect of focal length error on the end mismatch of the final panorama, and proposed a new scheme for fine-tuning the focal length estimate to help eliminate the end mismatch. Errors arising

from tilting and rolling of the initial image are considered and allowed for to further tidy the final panorama. Finally a deskewing method is further used to eliminate minor remaining defects. Experiments show that our method can produce visually satisfactory results, as will be shown in the next chapter.

Chapter 6

Implementation and Examples

This chapter provides examples of panorama constructed by the methods presented in this thesis. Four sequences of images both from film and digital cameras are illustrated. A spherical panorama constructed from computer-generated images is also demonstrated.

6.1 Implementation

To test and verify the methods presented in this thesis, a prototype system for panorama building and viewing was implemented on a PC platform. The system is a multi-document and multi-view Windows application. One window allows the user to perform all the tasks of automatic pairwise registration including tools for image processing and interactive registration; a second window provides an interface for setting up the image sequence and generating the panorama image. The final window shows the 2D panorama image as well as allowing 3D viewing. An overview of the system in operation is shown in Figure 6.1.

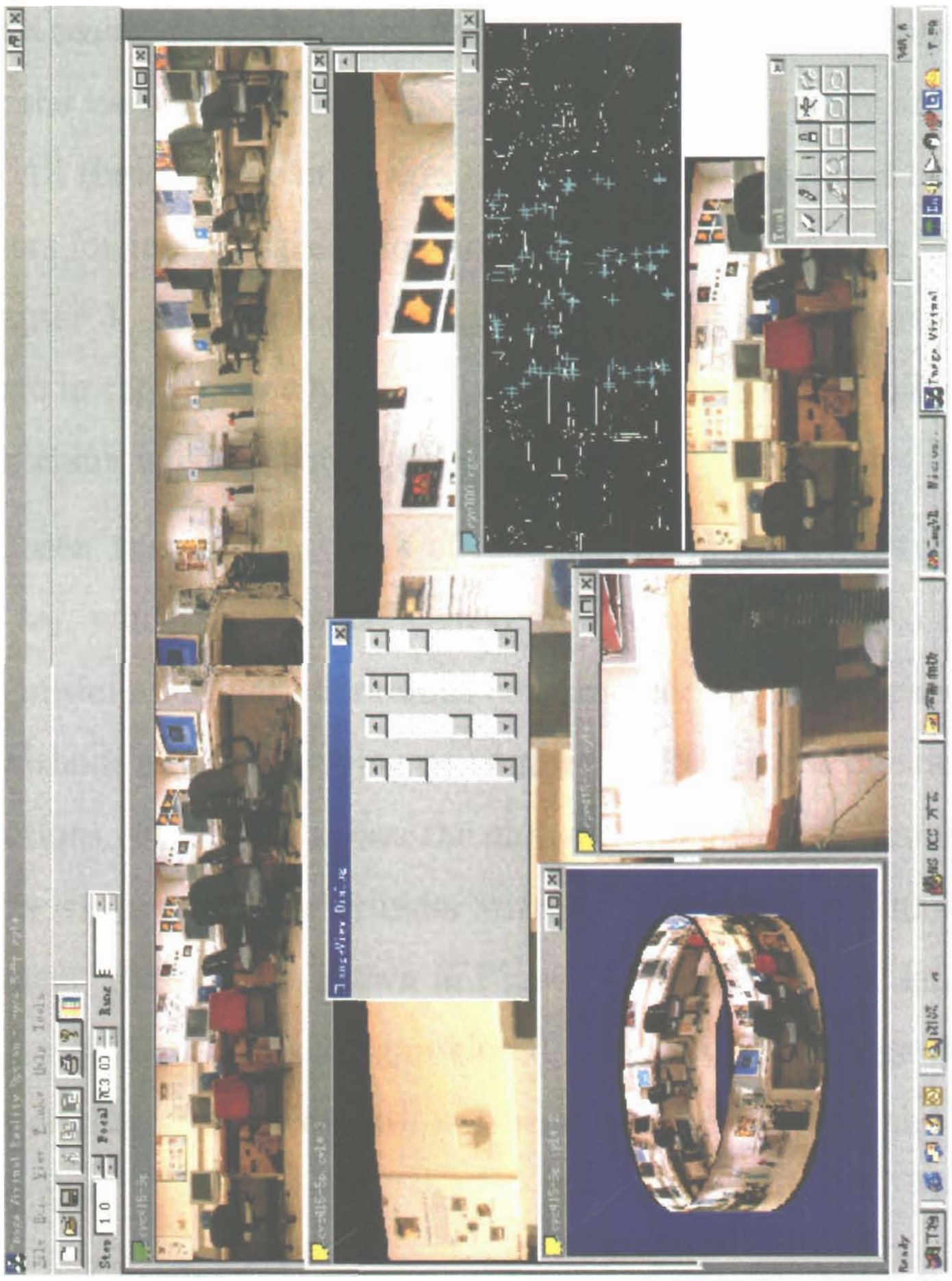


Figure 6.1: System Interface.

6.2 Cylindrical Panoramas

We now describe the test data used in this thesis for cylindrical panoramas. Figure 6.2 shows a sequence of 8 images scanned in from film photographs. Figures 6.3 and 6.4 show the tie-points detected in each pair of images—they are marked with '+' symbols. All these 8 pairs of images are successfully registered by our method. The parameters for image registration are set to be the same with that of experiments in Chapter 3. For pair lo_7 and lo_0, there is an object (a person) missing in the right image in the region which overlaps the left image. This offers a challenge to feature matching which is successfully handled.

It can be seen that sixteen pairs of matched tie-point are detected in image pair lo_7 and lo_0, which gives a sufficiently fine result. Figure 6.5 demonstrates the resulting initial and final panoramas constructed: (a) is the panorama image constructed without any final tidying corrections and (b) is the one built after final tidying corrections. Figure 6.6 shows the end gap when the panorama image after final tidying is wrapped onto a cylinder with focal length $f = 330.4$. After focal length refinement, the result is shown in Figure 6.7, and the focal length has been adjusted to $f = 323.16$, with the gap closed. To show the image quality before and after gap closing, intensities have *not* been blended between the last and first images in this case.

Figure 1.1 in Chapter 1 is another sequence of images scanned in from film photographs. A same set of parameters is used for registration between each adjacent image pair. Satisfactory results are obtained for all image pairs except pair $(3, 4)$, which contains a building having many similar feature points in the overlapping region, resulting in poor alignment. To deal with such a rare case, we may use interactive registration, which is also provided by the system. Figure 6.8

(a) shows the result of panorama construction without final tidying correction. The shape of the panorama makes it obvious that the first image must be tilted through a considerable angle. The final panorama image after correction is shown in Figure 6.8(b).

Gap closing is demonstrated in Figure 6.9: an overlap of last and first images results from mapping the panorama in Figure 1.2 onto a cylindrical surface. In Figure 6.9(a), $f = 330.40$. Figure 6.9(b) demonstrates gap closing after focal length refinement to $f = 334.36$ and Figure 6.9(c) shows the final result after all corrections including deskewing. Figure 6.10 shows the resulting cylindrical model.

Figures 6.11 and 6.12 show a sequence of images taken using a digital camera. There are 14 images in the sequence. In this example intensity differences are not apparent between adjacent images since they were all taken inside a room with constant illumination. All image pairs were able to be fine registered using our automatic two-stage registration approach. The final panorama and gap closing results are shown in Figure 6.13.

A sequence of 14 outdoor images taken using a digital camera is shown in Figures 6.14 and 6.15. Having a lot of texture in these images, many possible tie points are being found. It is obvious that excessive exposure differences are present between images of pairs $(2, 3)$ and $(4, 5)$. Although the fine registration program was invoked for these two pairs of images, the results from the first stage feature-based method were retained due to non-convergence of the gradient-based fine registration resulting from these significant intensity differences. The final panorama after final tidying correction is shown in Figure 6.16 (a). The results after end gap closing are illustrated in Figures 6.16(b) and (c).

6.3 Discussion

We now discuss the overall performance of our method. If we carefully observe Figures 6.3 and 6.4, we can see that the edge intensities are different in the same image when it appears in different pairs. This is because we have normalized the intensity in which the normalization coefficient is the same for each image in an image pair but varies between the pairs. When converting a grayscale edge image into a binary image, a maximum edge intensity value is first found in the minimal overlap region of the left image and is used as the normalization coefficient for both images. A pixel's value is set to 1 when its intensity value exceeds a certain percentage of this coefficient; otherwise it is set to 0. This normalization process helps to improve the robustness of feature detection by retaining the *common* features of interest in an image pair. If we normalize each image separately using its own maximum value over the whole of image, the features retained in the overlap region could be quite different for each image, since the image intensity may differ a lot in the non-overlap region as a whole between the images. This may cause many features in one image to have no counterparts in the other image, thus making matching more difficult.

When the actual overlap is much larger than the minimal overlap assumed, it is more likely to produce a coarse alignment only using the feature-based registration algorithm, because the features used do not cover the whole overlap region. If the user has some rough idea about the amount of overlap beforehand, this value can be used for the whole sequence. The default value of minimal overlap is set to 16%. Presumably we know the width of each image and the focal length, we can estimate the minimal overlap in this way: first we compute a width sum of all images in the sequence and the circumference with the focal length to be the radius; then we take the difference of these two values and divide it by the number of images.

Another simple way to set this value is to only use the number of images in the sequence. For instance, if there are less than 8 images in the sequence indicating a relatively small overlap, we can just use the default overlap fraction of 16%. If there are more than 8 but less than 14 images, a starting overlap value of 25% is more appropriate so that we include more features to obtain a finer resolution of feature-based registration. We may even use overlap values of 50% if more images are included in a sequence. Using a correspondence set covering the whole overlap region helps to achieve better registration result. However, using a larger minimal overlap extracts more features, resulting in more feature matches, and thus increases computational cost.

Setting of parameters for feature-based registration has already been discussed in Section 3.6 of Chapter 3. Further experiments show that our registration procedure is robust, and relatively insensitive to the choice of operational parameters. In practice, we set most of the parameters to default values. Only four parameters are left in the system interface for the user to adjust. One is T_g the threshold parameter for converting the gray scale edge image to binary black and white image. The user can adjust this value easily by observing the whether there are adequate significant edges remained. The other three parameters are related to the size of main structures in an image, which are derivative deviation σ, the length for determining curvature l and the size of the template window for matching correspondences s. The same parameter values were used for all the experiments shown in this thesis, i.e. $T_g = 0.2$, $\sigma = 2.5$, $l = 10$ and $s = 30$. When the gradient-based fine registration method is invoked, it makes use of the gradient images obtained from Canny edge detection in the first stage of feature-based registration. In this way the computational expense is held down.

Figures 6.2 and 1.1 are scanned in photographs from a film camera, while

Figures 6.11, 6.12, 6.14 and 6.15, are images from a digital camera. Generally the images from a digital camera have more consistent brightness between adjacent images compared with those scanned from film photographs, so we can obtain finer resolution at an earlier stage of whole registration process.

We are also considering some more objective ways to evaluate the quality of constructed panorama. Such as comparing our results with source images captured using a calibrated camera with turning table, or using a panorama camera, etc.

Tests show that our method is fast. The timing of pairwise registration in a Pentium III 500 MHz PC is about 2∼5 seconds for a typical image pair shown in this thesis, with the image size ranging between 320∼390 for width and 240∼260 for height in pixel. The feature-based registration needs about 2∼3 seconds, in which Canny edge detection takes about 1.5 seconds. Once invoked, the fine registration is efficient since the gradient image is already available from the previous feature extraction operation. Hence, one or two more seconds are needed for 3- or 5-parameter fine registration models, respectively. Since the 4-parameter feature-based model includes all the camera constraints in our case, a small number of tie points are sufficient for a reasonably good overall alignment. Canny edge detection is the step which overall takes most time. Typically, about 2 minutes is used for constructing a panorama consisting of a dozen images shown in this thesis, among which about 45 seconds is used by computer processing, the rest is by manual arrangement.

6.4 Spherical Panoramas

An example of constructing a spherical panorama from computer generated images is shown in Figure 6.17. There are three sets of image sequences which cover a

closed space, with 7 images in each set. The small green square marks the center of an image while the red lines show the neighboring relations of each image. Using the methods in Section 5.4, a spherical panorama was constructed. The result is shown in Figure 6.18.

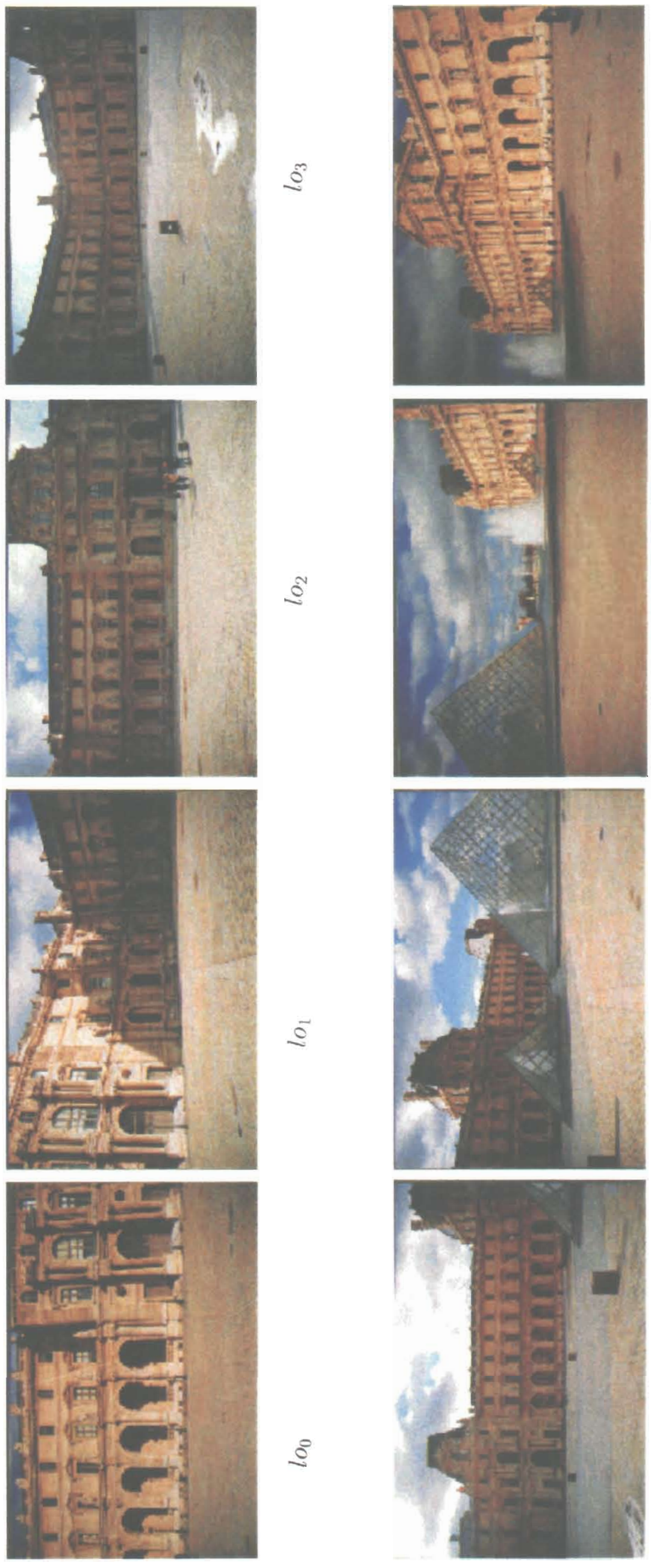
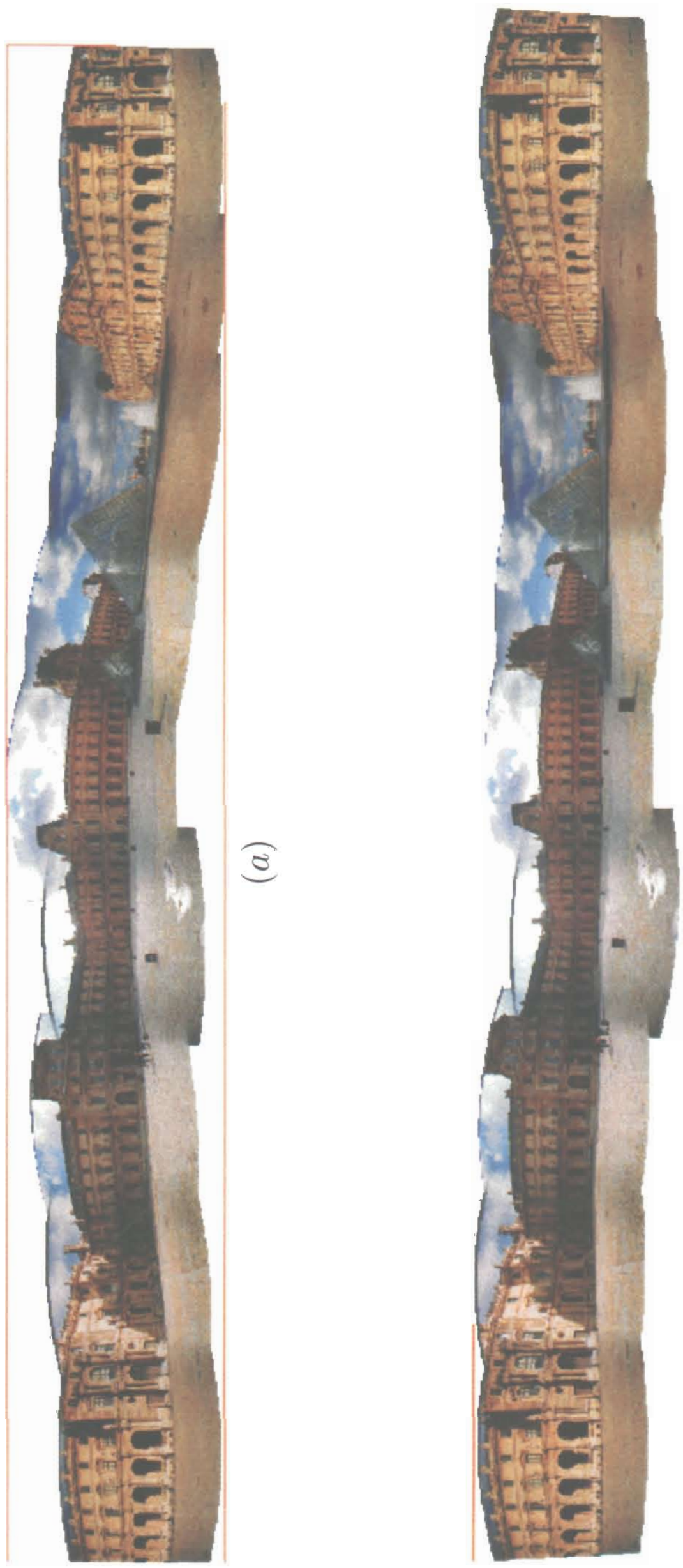


Figure 6.2: A sequence of images: $lo_0 \sim lo_7$.

Figure 6.3: Tie-points detected on image pairs.

Figure 6.4: Tie-points detected on image pairs.



(a) (b)

Figure 6.5: Initial (a) and final (b) cylindrical panoramas.

Figure 6.6: End gap ($f = 330.4$)

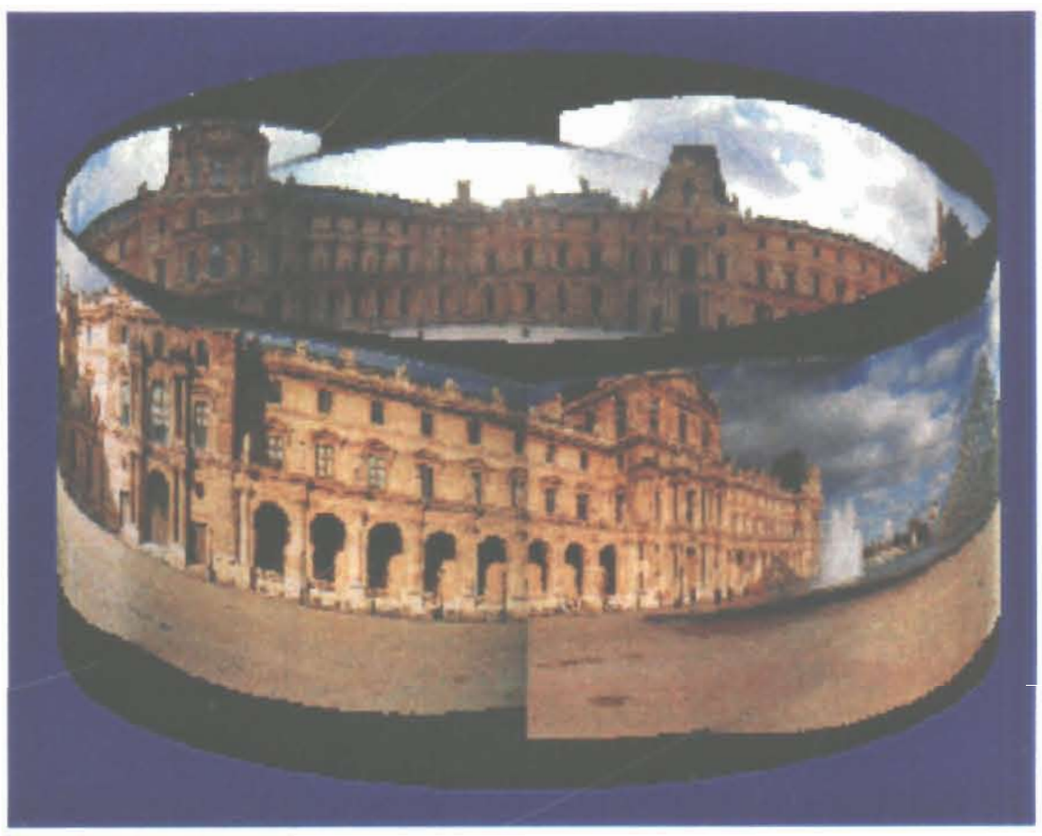


Figure 6.7: Gap closed($f = 323.16$).

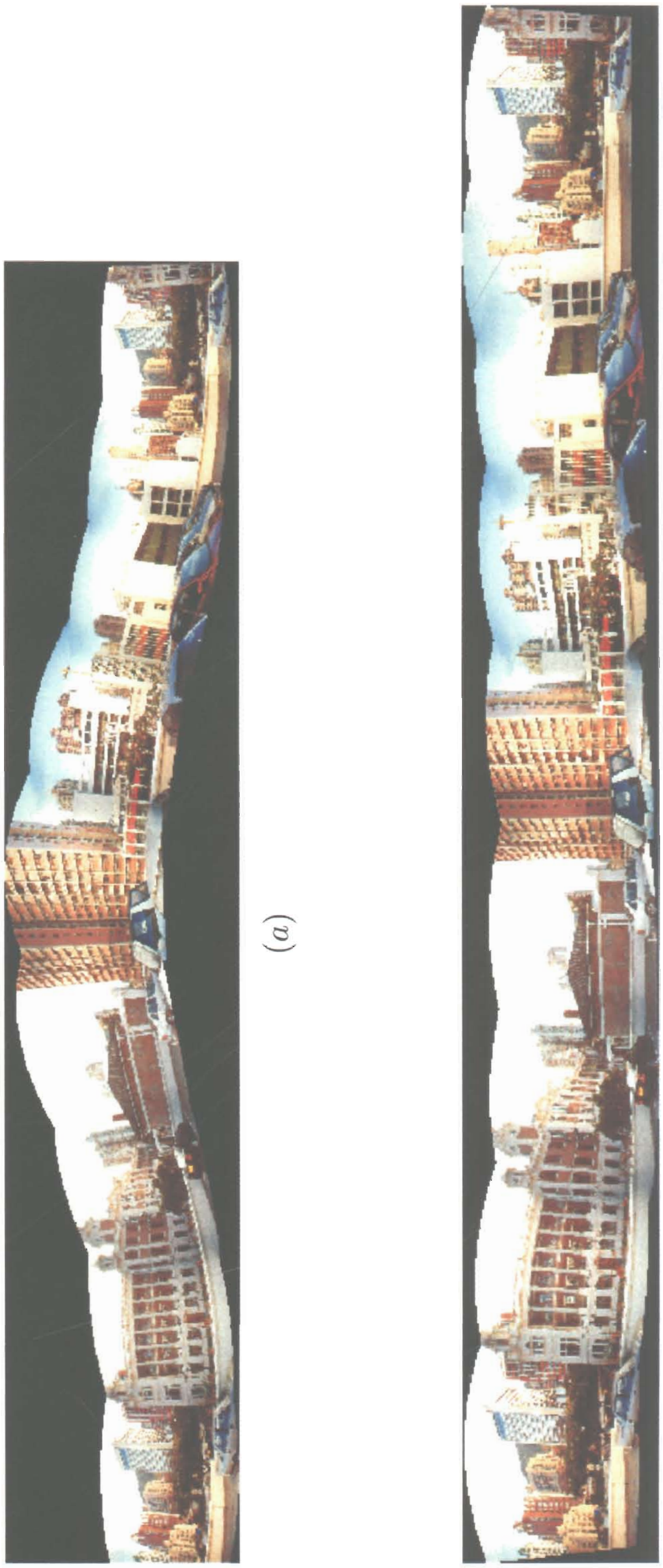


Figure 6.8: Initial (a) and final (b) cylindrical panoramas.

(a) (b) (c)

Figure 6.9: Gap closing (a) mismatch at the end; (b) focal length refinement; (c) further adjustment.

Figure 6.10: Mismatch closed in cylinder model.

$scysl_3$

$scysl_7$

$scysl_2$

$scysl_6$

$scysl_1$

$scysl_5$

$scysl_0$

$scysl_4$

Figure 6.11: A sequence of images: $scysl_0 \sim scysl_0$.

$scysl_{11}$ $scysl_{10}$ $scysl_9$ $scysl_8$ $scysl_{13}$ $scysl_{12}$

Figure 6.12: A sequence of images: $scysl_8 \sim scysl_{13}$.

(a)

(c)

(b)

Figure 6.13: (a) cylindrical panoramas, (b) end gap and (c) gap closed.

gdm_0 gdm_1 gdm_2 gdm_3

gdm_4 gdm_5 gdm_6 gdm_7

Figure 6.14: A sequence of images: $gdm_0 \sim gdm_7$.

gdm_8 gdm_9 gdm_{10} gdm_{11}

gdm_{12}

Figure 6.15: A sequence of images: $gdm_8 \sim gdm_{12}$.

(a)

(b)

(c)

Figure 6.16: (a) cylindrical panoramas, (b) end gap $f = 358$, and (c) gap closed $f' = 353.35$.

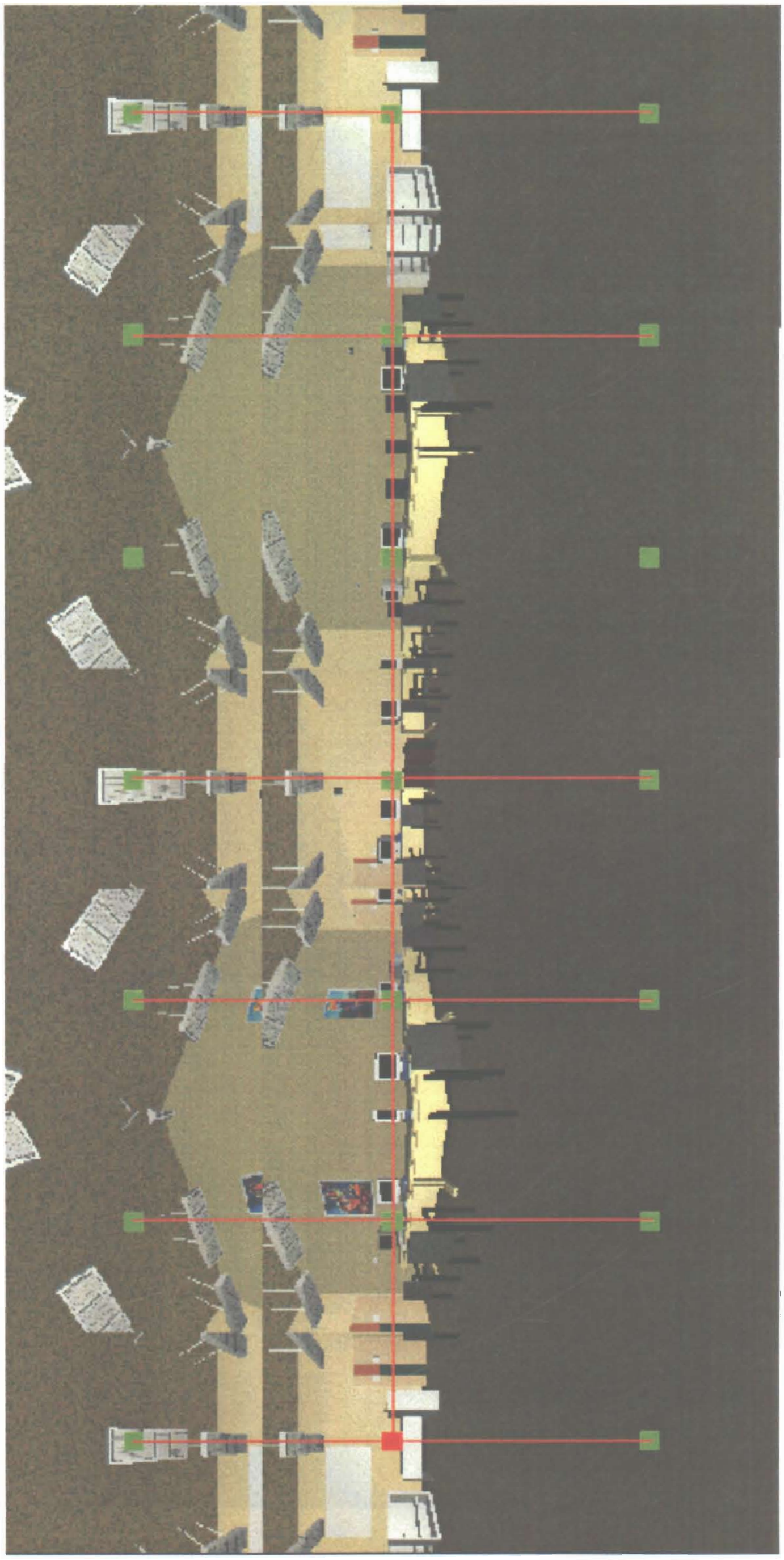


Figure 6.17: Three sets of image sequences.

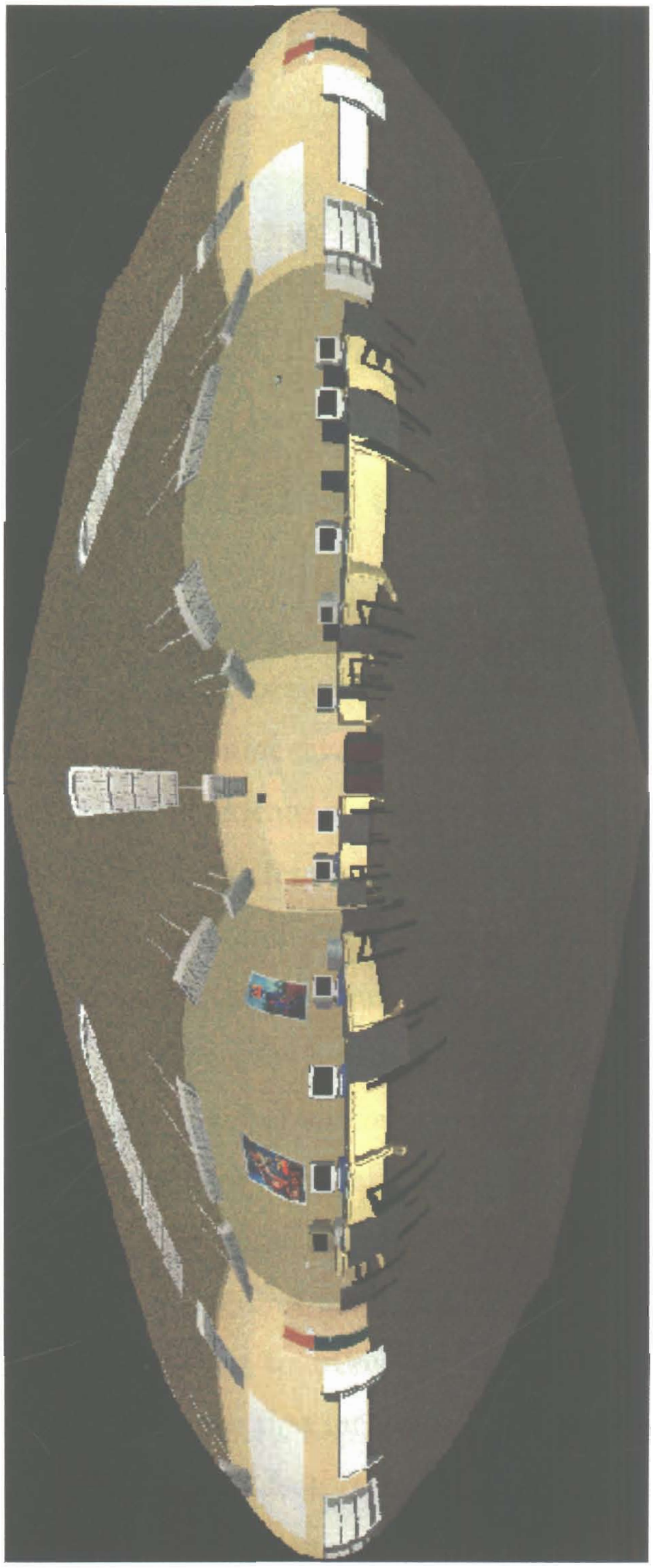


Figure 6.18: Spherical panorama.

Chapter 7

Conclusions and Future Work

A panoramic image is a compact representation of a large field of view which can be used to provide users with an immersive view of an environment. It is a key tool for providing a virtual reality experience of a complex scene. It also has other wide visualization applications across the Internet on the World Wide Web. Among the various forms of panorama, the cylindrical model is the most commonly adopted, in which a collection of images is used to render the scene while supporting circular camera motion. The source images for a panorama can be obtained using special panoramic cameras or with the aid of special equipment like a turntable, or just using a hand-held ordinary camera. The latter approach eases the restrictions of image acquisition for a non-specialised user in that it tolerates moderate camera tilting and rolling. However, it requires more effective methods for constructing a high quality panoramic image than traditional mosaicing techniques. This thesis has presented tools for this purpose that include a robust pair-wise image registration method using a combination of feature-based and gradient-based techniques, and also techniques for panorama tidying to correct various minor errors in order to generate a visually satisfactory final result.

7.1 Synopsis

In this thesis, I have presented a method for building a full panorama of a 3D scene from uncalibrated photographs taken with an ordinary hand-held camera located at an approximately fixed location. The method is automatic and imposes no stringent requirements on the photographer.

The method is a two-step image registration process using a feature-based method for initial registration, and a gradient-based method for further fine registration if needed. The key to the first step is a robust procedure for finding reliable feature correspondences between pairs of adjacent images. To do this, an improved algorithm for high curvature point detection is used for feature extraction, and a gradient and a shape based metric are used for feature matching.

I have also shown how to model the perspective transformation between two adjacent images, and have given an iterative technique using a sequence of linear steps for computing this perspective transformation by minimizing an error function.

To improve the result of feature-based image registration, a gradient based fine registration is invoked when there are too few features detected, or some when certain mismatched features are not excluded leading to poor initial registration. The latter problem poses a serious problem to the least square optimization process used in the feature registration approach and is a major cause of large residual errors. A new 5-parameter model has been proposed for fine registration which can generate better results than the existing 3-parameter model and is computational more efficient than the existing 8-parameter model. An analysis of suitable choice of smoothing factor in fine registration is presented to enlarge the effective registration scope.

I have described how deficiencies in the final panorama, caused by treating panorama building as a series of local registration problems rather than a global problem, can be overcome by a sequence of correction techniques. In particular, I discussed cylindrical warping methods that allow minor tilting and rolling of the images, and explained how to correct deficiencies in the panorama caused by assuming zero tilting and rolling of the first image of the sequence.

Finally, the effects of errors in focal length estimation on panorama closing were investigated theoretically, and it was found that the length of the composite panorama is more sensitive to focal length error than to errors in local registration. Based on this observation, a new approach for gap closing by iteratively adjusting the focal length and *panning* angles accordingly was proposed.

A prototype system has been developed to verify the theoretical approaches presented in the thesis. I have tested the methods with photographs of a number of different 3D scenes, and the overall system has yielded visually satisfactory results. The most time-consuming step is pair-wise image registration. However, since a relatively small number of features are used for performing an initial registration, and also because the image processing results from the first step are re-used by the second fine registration step, the overall speed of our method is reasonably fast.

Automatic image stitching tools are provided by various pieces of commercial software such as LivePicture (MGI)'s Photovista, Apple's QuickTime VR, and IBM's PanoramIX. We have tried using them on various images scanned from film photographs and digital camera pictures, and I have found that their automatic registration capabilities are not very reliable when there is a small overlap or large brightness differences present between adjacent images. Two examples are given below. Figure 7.1 is the output of the Photovista automatic stitching function for the image sequence shown in Figure 6.2, while the panorama image produced by

our method is shown in Figure 6.5(b). We can see in Figure 7.1 that only two pairs of images are aligned by Photovista, while our method did successfully register all pairs. Figure 7.2 shows another example of the use of Photovista, using the digital camera images given in Figures 6.14 and 6.15; our result is shown in Figure 6.16 (a). For this set of digital photographs, there are four image pairs that Photovista obviously misregistered.

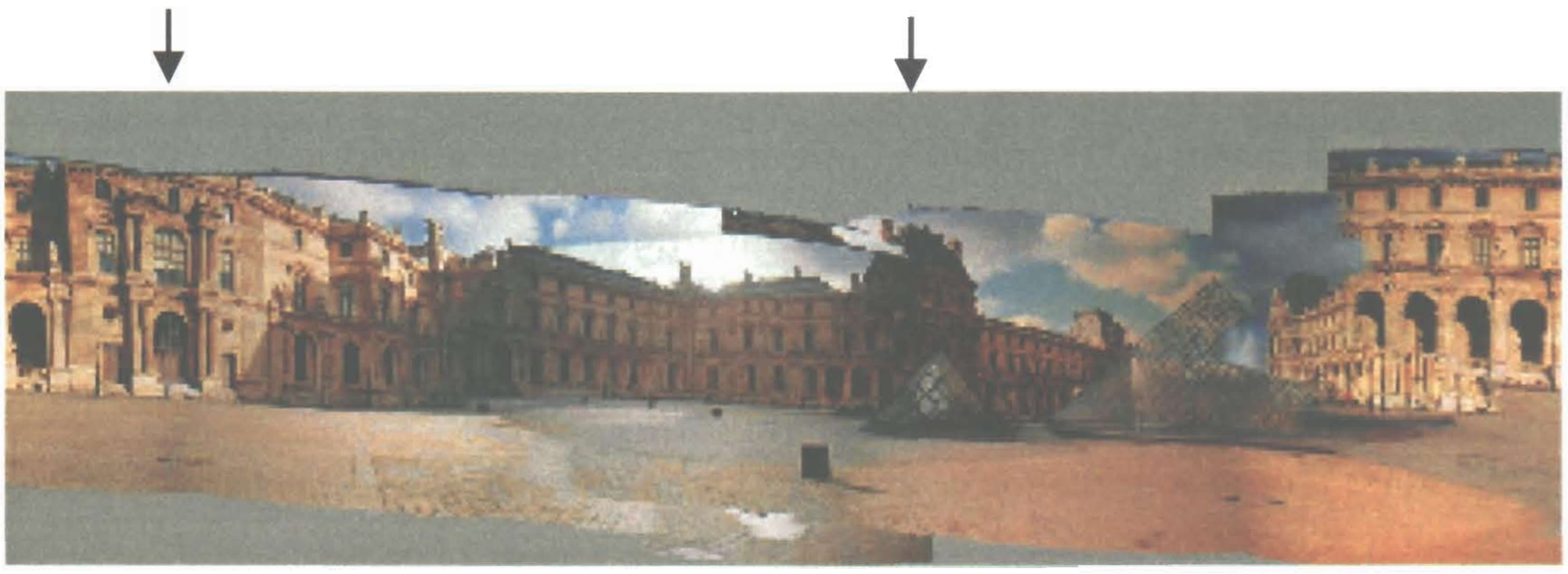


Figure 7.1: Result from Photovista for scanned in film photographs $lo_0 \sim lo_7$

Figure 7.2: Result from Photovista for digital camera image sequence $gdm_0 \sim gdm_{12}$

7.2 Future Work and Discussions

In this last section I will discuss some limitations of the work presented in this thesis and future research directions.

7.2.1 More Flexible Model

In this work, I have assumed that all the images in a sequence have the same fixed focal length; the image alignment solution depends on this. A more flexible model should allow variations of focal length, that is zooming of the camera. So an interesting problem is to consider the presence of changing focal length as well as of large perspective distortions.

Another assumption used for panorama construction in this thesis is that all images are taken from a nearly fixed viewing position. This means we may use a perspective transformation to approximately model the relation between adjacent images used to construct the panoramic image. A perspective transformation is only correct when the images are taken from a common viewpoint while rotating a camera (or when the scene is planar with arbitrary camera motion). If these conditions are not satisfied, then the resulting panorama will not be physically correct. Although alignment of images can be modeled by a polynomial transformation when it cannot be modeled by a perspective transformation, this method does not give information on how to decide the transformation between the images outside the overlap region. It might be possible, however, to combine a global registration that uses all features in the whole sequence of images together with some local adjustment methods.

7.2.2 Correction of Camera Lens Distortions

To make it easy for users to acquire input images, intricate calibration procedures should be avoided or restricted to a minimum. Camera intrinsic parameters such as focal length and radial distortion coefficients may be found by such procedures. In this work, the focal length is estimated using the rotation model and perspective

transformations, and refined using the composite panorama length. To simplify the problem, I have assumed the radial distortion of camera is negligible. However, given a camera with a low quality lens, lines near the image border can be distorted, and brightness variations can occur there too. For example, for the images in Figure 6.2, the intensity is lower in the four corners than in the central area, and lines are bent near the edges of the image. These are typical phenomena produced by radial distortion with low-scale lenses. There are many approaches to finding radial distortion coefficients [Kang99, Sawhney99], but efficient methods are still worthy of exploration. To correct intensity distortions, a proper blending scheme is needed which can smooth the intensity across the whole panoramic image and remove unwanted variations. This is another interesting area to study, either to produce an automatic method, or to devise some interactive tools which can correct gray levels in any area of interest in an image and at the same time blend the border smoothly with its surroundings.

7.2.3 Global Solutions for Eliminating End Seams

The end-seam elimination methods described in this thesis adjust the pair-wise registration locally. A better approach in principle is a global registration solution that makes use of features extracted from all images of the circular sequence simultaneously. By using a global reference frame to represent the features' coordinates in 3D camera reference frames while at the same time imposing the closure constraints in 3D form, it is possible to formulate and simplify the problem to provide a solution at reasonable cost.

7.2.4 Multiple Image Registration for Spherical Panorama Construction

Using a cylindrical panoramic image has the drawback that it is unable to include parts of the scene above and below a given strip. This deficiency can be overcome by using a spherical panorama. To build a spherical panorama from images taken using an ordinary camera, several sequences of images each with a different fixed tilting angles can be taken to cover a closed space. Since with this approach there will be at least four adjacent images meeting in each overlap region, multiple image registration must be performed. Since manually registrating multiple images is almost impossible to provide a satisfactory result, automatic registration is an indispensable step in constructing a spherical panorama. So efficient multiple image registration algorithms are another avenue worthy of exploration, considering its wide potential applications.

7.2.5 Optimum Number of Images

Another practical issue that needs further investigation is how best to choose the number of images taken. Fewer images make the panorama building process more economical, but will reduce the resolution of the panorama. Including more images will increase the cost of construction in both speed and storage. The number of images can also affect algorithmic robustness; For instance, if there are too few images in a sequence, the overlap regions of adjacent images will be too small to have enough features included, thus causing failure of the registration algorithm, or at least, poor registration. To decide the optimum number of images, more experimental work should be carried out.

7.2.6 The Future

Now that full view panoramas at a single viewpoint can be successfully constructed from images from a hand-held camera, a source of data for studying further issues in image-based virtual reality systems is available, leading to an area with tremendous potential for exploration. By easing the restrictions on source images used for building panorama images, this work will make more users interested in panoramic photography, stimulate the demand for constructing visual scenes and promote further research in the area.

Bibliography

[Alvarez00] L. Alvarez, J. Weichert and J. Sanchez, Reliable Estimation of Dense Optical Flow Fields with Large Displacements, *International Journal of Computer vision*, Vol. 39(1), PP.41-56, 2000.

[Anandan89] P. Anandan, A Computational framework and an algorithm for the measurement of vidual motion, *International Journal of Computer Vision*, Vol. 2, PP. 283-310, 1989.

[Aschwanden93] P. Aschwanden and W. Guggenbuehl, Experimental Results From a Comparative Study on Correlation-Type Registration Algorithms, *Robust Computer Vision*, Foerstner and Ruwiedel, eds., pp. 268-289. Wichmann, 1993.

[Aubert99] G. Aubert, R. Deriche, and P. Kornprobert, Computing Optical Flow Via Variational Techniques. *SIAM Journal on Applied Mathematics*, Vol. 60(1), pp. 156-182, 1999.

[Barron94] J.L. Barron, D.J. Fleet, and S.S. Beauchemin, Sytems and Experiment, Performance of Optical Flow Techniques, *International Journal of Computer Vision*, Vol.12(1),PP. 43-77, 1994

[Bao99] P. Bao, and D. Xu, Complex Wavelet-Based Image Mosaics Using Edge-Preserving Visual Perception Modelling, *Computer Vision*, Vol. 23(3), pp. 309-321, June 1999.

[Brown92] G. Brown, A Survey of Image Registration Techniques. *ACM Computing Surveys*, Vol. 24(4), pp. 325-376, December 1992.

[Bergen92] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani, Hierarchical Model-Based Motion Estimation. In *Proc. European Conference on Computer Vision*, pp. 237-252, Santa Margarita Ligure, May, 1992.

[Bhat98] D. N. Bhat and S. K. Nayer, Ordinal Measures for Image Correspondence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.20(4), pp. 415-423, April, 1998.

[Bonet98] J.S. De Bonet and A. Chao, Structure-Driven SAR Image Registration. In *Proceedings of SPIE 1998*, Vol.3370, pp.109-117, 1998

[Burt83] P.J. Burt and E.H. Adelson, A Multiresolution Spline With Application to Image Mosaics. *ACM Transactions on Graphics*, Vol.2(4), pp. 217-236, Oct. 1983.

[Canny86] J. Canny, A Computational Approach to Edge Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.8(6), pp. 679-698, Nov. 1986.

[Castro87] E. De Castro and C. Morandi, Registration of translated and rotated images using finite Fourier Transforms, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 9(5), pp. 700-703, 1987.

[Chang97] N.L. Chang A. Zakhor. View generation for three-dimensional scenes from video sequences. *IEEE Transactions on Image Processing*, 6(4), pp.584-598, 1997.

[Chen93] S.E. Chen L. Williams. View interpolation for image synthesis. In *Proc. of ACM SIGGRAPH,* 1993.

[Chen95] S.E. Chen. QuickTime VR - An Image-Based Approach to Virtual Environment Navigation. *Computer Graphics (SIGGRAPH'95)*, pp. 29-38, August 1995.

[Chen97] Qian Chen and Gerard Medioni. Image Synthesis From A Sparse Set Of Views. *Proceedings of Visualization '97*, page 269-275, 1997.

[Chen00] H. Chen, W. Wang and R. Martin, Building Panoramas from Photographs Taken with an Uncalibrated Hand-Held Camera. *Proc. Of Vision, Modeling and Visualization 2000*, Saarbrucken, Germany, pp. 221-230, 2000.

[Chiang93] J. Y. Chiang and B. J. Sullivan, Coincident Bit Counting—A New Criterion for Image Registration. *IEEE Trans. On Medical Imaging*, Vol. 12(1), March 1993.

[Coorg98] S. Coorg, N. Master and S. Teller. Acquisition of a large pose-mosaic dataset. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

[Debevec96] P.E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. *Computer Graphics (SIGGRAPH'96)* pp. 11-20 August 1996.

[Deriche93] R. Deriche and T. Blaszka. Recovering and characterizing image features using an efficient model based approach. In *Proc. Of International Conference on Computer Vision and Pattern Recognition* , pp.530-535, New York, June 1993.

[Davatzikos96] C. Davatzikos, J. L. Prince and R. N. Bryan. Image registration based on boundary mapping. *IEEE Transaction on Medical Imaging*, Vol.5(1), pp.112-116, Feb. 1996.

[Dhond89] U.R. Dhond and J. K. Aggarwal, Structure from Stereo—A Review. *IEEE Trans. Syst. Man Cybernetics*, Vol.19(6) pp.1489-1510, Nov./Dec. 1989.

[Enroute] , Enroute Imaging's Quickstich *http:/www.enroute.com*

[Faugeras00] Faugeras, O.; Long Quan; Strum, P. Self-calibration of a 1D projective camera and its application to the self-calibration of a 2D projective camera, *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp.1179 - 1185 Vol.22(10) Oct. 2000

[Faugeras93] O. Faugeras. *Three Dimensional Computer Vision.* MIT press, 1993.

[Ghosh88] , Sanjib K. Ghosh, *Analytical Photogrammetry*, second edition. Pergamon. 1988.

[Gortlet96] Steven J. Gortlet, Radek Grzeszczuk, Richard Szeliski and Michael F. Cohen. The Lumigraph. *Computer Graphics (SIGGRAPH '96)*, page 43-54, 1996.

[Gonzalez93] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Addison-Wesley, 1993.

[Goshtasby85] A. Goshtasby, Template Matching in Rotated Images. *IEEE Trans. Pattern Recognition and Machine Intelligence*, Vol. 7(3) pp. 338-344,1985.

[Gracias00] Nuno Gracias, Jos* Santos-Victor,1 Underwater Video Mosaics as Visual Navigation Maps, *Computer Vision and Image Understanding* , pp. 66-91, Vol. 79(1), July 2000

[Greene86] N. Greene, Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications.* Vol. 6(11) pp. 21-29. 1986.

[Guo92] Z. Guo, and R. W. Hall, Fast Fully Parallel Thinning Algorithm. *Computer vision, graphics, and image processing*, vol. 55(3),pp. 317 328, 1992.

[Gmstekin98] , S. Gmstekin R.W. Hall. Image registration and mosaicing using a self calibrating camera. In *Proc. of IEEE Int. Conf. on Image Processing*, 1998.

[Haralick93] R. Haralick and L. Shapiro, *Computer and Robot Vision Volume 2.* Reading, MA: Addison-Wesley, 1993.

[Harris88] C. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vission Conf.* pp. 189-192, 1988

[Hartley94] R. I. Hartly and R. Gupta. Linear Pushroom Camera. *Proc. Third Eropean Conf. Computer Vision*, J.O.Eklundh, ed., pp.555-566, May 1994.

[Hartley94-1] R. I. Hartly. Self-calibration from multiple views with a rotating camera, *in Third European Conference on Computer Vision (ECCV'94).* Vol.1, Stochholm, May 1994. Springer-Verlag, Berlin/New York, pp. 471-478.

[Horn86] Berthold Klaus Paul Horn, *Robot Vision.* The MIT Press Cambridge, Massachusetts, London England. 1986.

[Horn81] B.K.P Horn and B.G. Schunck, Determine Optical Flow. *Artificial Intelligence*, Vol. 17, pp. 185-203, 1981.

[Hotmie] IBM, Hotmedia, *http:/www4.ibm.com/software/net.media*

[Irani95] M. Irani, S. Hsu, and P. Anandan, Video compression using mosaic representations. *Signal Processing: Image Communication, special issue on Coding Techniques for Low Bit-rate Video*, Vol. 7, No. 4-6, pp. 529-552, November 1995.

[Irani96] M. Irani P. Anandan J. Bergen R. Kumar S. Hsu. Mosaic representations of video sequences and their applications. *Signal Processing: Image Communication, special issue on Image and Video Semantics: Processing, Analysis, and Application*, 8(4), May 1996.

[IPIX] Interactive Pictures' IPIXTM Multimedia Builder *http:/www.ipix.com/*

[Irani98a] M. Irani P. Anandan. Video indexing based on mosaic representations. *Proceedings of the IEEE*, Vol.86(5) pp.905-921, May, 1998.

[Irani98] M. Irani P. Anandan. Robust multi-sensor image alignment. In *Proc. of IEEE International Conf. on Computer Vision*, India, Jan 1998.

[John86] C. John, A Computational Approach to Edge Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-8, No.6, pp. 679-698, Nov. 1986.

[Kanade97] T. Kanade P.W. Rander P.J. Narayaman. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Trans. on Multimedia,* Vol.4(1) pp.34-47, 1997.

[Kang98] Sing Bing Kang, Pavan K. Desikan, Virtual Navigation of Complex Scenes using Clusters of Cylindrical Panoramic images, *Graphics Interface'98*, pp.223-232, 1998.

[Kang99] Sing Bing Kang, Richard Weiss, Characterization of Errors in Compositing Panoramic Images, *Computer Vision and Image Understanding* Vol. 73(2) pp. 269-280 , February 1999 .

[Kuglin75] C.D. Kuglin and D. C. Hines, The Phase-correlation Image Alignment method. In *Proceedings of IEEE international conference on Cybernetics and Society*, New York, PP163-165. 1975.

[Kruger98] S. Kruger and A. Calway, Image Registration using Multiresolution Frequency Domain Correlation, In *British Machine Vision Conference*, British Machine and Vision Association, pp.316-325, September 1998.

[Levo96] Marc Levoy and Pat Hanrahan. Light Field Rendering. *Computer Graphics (SIGGRAPH '96)*, page 31-42, 1996.

[Meehan90] J. Meehan, *Panoramic Photography*, Watson-Guptill, 1990.

[McMillan95] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *Computer Graphics (SIGGRAPH'95)*, pages 39-46, August 1995.

[Milgram75] D.L. Milgram, Computer Methods for Creating Photomosaics, *IEEE Transactions in Computers*, Vol.C-24, pp.1113-1119, 1975.

[Milgram77] D.L. Milgram, Adaptive Techniques for Photo Mosaicing, *IEEE Transactions in Computers*, Vol. C-26, pp. 1175-1180, 1977.

[Maurer98] C. R., Maurer Jr., R. J. Maciunas and M. Fitzpatrick, Registration of Head CT Images to Physical Space Using a Weighted Combination of Points and Surfaces. *IEEE Trans. On Medical Imaging*, Vol.17(5), PP. 753-762, Oct. 1998.

[Nage86] H. H. Nagel and W. Enkelmann, An Investigation of Smoothness Constraints for the Estimation of Displacement Vector Fields From Image Sequences, *IEEE Trans. Pattern Analysis And Machine Intelligence*, Vol.8(5), September, 1986.

[Nage87] H. H. Nagel, On the Estimation of Optical Flow: Relations Between Different Approaches and Some New Results, *Artificial Intelligence*, Vol.33, PP. 299-324, 1987.

[Nayar97] S.K. Nayar, Catadioptric Omnidirectional Camera. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)* San Juan, Puerto Rico, pp. 482-488, June,1997.

[Li95] H. Li, B.S. Manjunath, and S.K. Mitra, A Contour-Based Approach to Multisensor Image Registration. *IEEE Transaction on Image Processing* Vol. 4(3), pp. 320-334, March 1995.

[Onoe98] Y. Onoe K. Yamazawa H. Takemura N. Yokoya. Tele-presence by real-time view-dependent image generation from omnidirectional video streams. *IEEE Trans. on Computer Vision and Image Understanding,* Vol.71(2) pp.154-165, Aug 1998.

[Panavue] Panavue's Visual Sticher, *http://www.panavue.com/*

[Photov] MGI, Photovista, *http:/www.mgi.com*

[Pratt74] K. Pratt, Correlation Techniques of Image Registration, *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-10. pp. 353-358, May, 1974.

[Pratt91] W. Pratt. *Digital Image Processing*, New York: Wiley, 1991.

[Peleg81] S. Peleg, Elimination of Seams from Photomosaics. *Computer Graphics and Image Processing*, Vol.16, pp. 90-94,1981.

[Peleg97] S. Peleg and J. Herman, Panoramic Mosaics by Manifold Projection, In *Proc. CVPR'97*, pp. 338-343, June 1997.

[Peleg00] S. Peleg, B. Rousso, A. Rav-Acha and A.Zomet. Mosaicing on adaptive manifolds. In *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol: 22 (10) pp: 1144 - 1154 ,Oct. 2000.

[pictureworks] PictureWorks' Spin Panorama *http:/www.pictureworks.com/*

[Press92] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge, England, 1992.

[Qickvr] QuickTime VR,*http:/www.quicktime/qtvr*

[Rademacher98] P. Rademacher and G. Bishop. Multiple-Center-of-Projection Images. *Computer Graphics (SIGGRAPH'98)*, pages 199-206, July 1998.

[Reddy96] B.S. Reddy and B.N. Chatterji, An FFT-based technique for translation, rotation, and scale-invariant image registration, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 5(8), pp. 1266-1271, 1996.

[Rousso97] B. Rousso, S. Peleg and I. Finci. Generalized Panoramic Mosaics, *Proc. DARPA Image Understanding Workshop'97*, pp. 255-260, May 1997.

[Satoshi87] S. Satoshi and A. Keiichi, Analysis of Template Matching Thinning Algorithm. *Pattern Recognition*, Vol. 20(3), pp.297 307, 1987.

[Sawhney99] H.S. Sawhney, and R. Kumar, True Multi-Image Alignment and ites Application to Mosaicing and Lens Distortion Correction, *IEEE Trans. Pattern Analysis and Machining Intelligence*, Vol. 21(3), March 1999.

[Shiren89] Y. Shiren, L. Li, and G. Peng, Two-Dimensional Seam-Point Searching in Digitial Image Mosaicing. *Photogrammetric Engineering and Remote Sensing*, Vol. 55, No:1, pp. 49-53, 1989.

[Shum98] H.Y. Shum M. Han R. Szeliski. Interactive construction of 3d models from panoramic mosaics. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, June 1998.

[Shum00] H. Y. Shum, and R. Szeliski, Construction of Panoramic Image Mosaics with Global and Local Alignment, *International Journal of Computer Vision*, Vol. 36(2), pp.101-130, February 2000.

[Stockman82] G. C. Stockman and S. Kopstein, Matching Images to Models for Registration and Object Detection via Clustering. *IEEE Trans. On Pattern Recognition and Machine Intelligence.* Vol. 4, pp.229-241, 1982.

[Szeliski96] R. Szeliski, Video mosaics for virtual environments , *IEEE Computer Graphics and Applications*, Vol.16(2),pp. 22 - 30, March 1996

[Szeliski97] R. Szeliski, Spline-Based Image Registration. *International Journal of Computer Vision*, Vol. 22(3), pp. 199-218, March/April 1997.

[SzelS97] R. Szeliski and H. Y. Shum, Creating Full View Panoramic Image Mosaics and Environment Maps. *Computer Graphics (SIGGRAPH'97)*, pp. 251-258, August 1997.

[Stein95] G. Stein, Accurate internal camera calibration using rotation, with analysis of sources of error. *in Fifth International Conference on Computer Vision (ICCV'95)*. Cambridge MA, pp 230-236, June 1995.

[Super95] B.J. Super and A. C. Bovik, Shape from Texture Using Local Spectral Moments, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 17(4), pp. 333-343, 1995.

[Singh92] M. Singh, R. R. Brechner, and V. W. Henerson, Neuromagnetic Localization Using Magnetic Resonance Imagaes. *IEEE. Trans. Med. Imag.*, Vol.11(1), pp. 129-134, 1992.

[Smoothmove] Infinite Pictures' SmoothMoveTM, *http:/www.smoothmove.com*

[Terran] Terran Electrifier, *http:/www.terran.com/index.html*

[Terzopoulos86] D., Terzopoulos, Regularization of Inverse Visual Problems Involving Discontinuties, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.8, PP. 413-424, 1986.

[Thevenaz98] P. Thevenaz, U.E. Ruttimann and M. Unser. A pyramid approach to subpixel registration based on intensity. *IEEE Trans. Image Processing*, Vol. 7(1), pp: 27 - 41 Jan. 1998

[Tretiak84] O. Tretiak and L.Pastor, Velocity Estimation from Image Sequences with Second Order Differential Operators, *P*ro. 7th ICPR, pp.16-19, Montreal, 1983.

[Ventura90] A. Ventura, A. Rampini, and R. Schettini, Image Registration by Recognition of Corresponding Structures. *IEEE Trans. Geosci. Remote Sensing,* Vol. 28, pp. 305-314, May 1990.

[Verri89] A. Verri and T. Poggio, Motion Field and Optical Flow: Qualitative Properties, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.11(5) PP. 490-498, 1989.

[VRML] Web3D Consortium, *http://www.vrml.org/*

[Wood97] Daniel N. Wood, Adam Finkelstein, John F. Hughes, Craig E., Thayer and David H. Salesin, Multiperspective panoramas for cel animation *Computer Graphics (SIGGRAPH'97)*, pp. 243 - 250, August 1997.

[Xiong97] Y. Xiong and K. Turkowski. Creating Image-Based VR Using a Self-Calibrating Fisheye Lens. In *Conference on Computer Vision and Pattern Recognition (CVPR'97)*, San Juan, Puerto Rico page 237-243, 1997.

[Xiong98] Y. Xiong and K. Turkowski, Registration, Calibration and Blending in Creating High Quality Panoramas, In *Fourth IEEE Workshop on Applications of Computer Vision*, Princeton, New Jersey, October 19-21, 1998, pp. 69-74.

[Yang99] Z. Yang and F.S.Cohen, Image registration and object recognition using affine invariants and convex hulls, *IEEE Trans. on Image Processing* , vol. 8(7), pp. 934 - 946,

[Zheng92] J.Y. Zheng and S.Tsuji. Panaromic representation for route recognition by a mobile robot. *International Journal of Computer Vision*, 9(1):56-76, 1992.

[Zheng93] Q. Zheng and R. Chellappa, A Computational Vision Approach to Image Registration. *IEEE Trans. Image Processing*, Vol.2(3), pp. 311-326, July 1993.

[Zheng99] J.Y. Zheng and S.Tsuji. Generating dynamic projection images for scene representation and understanding. *Computer Vision and Image Understanding*, Vol.72, pp 237-256, December 1998.

[Zhang00] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Analysis and Machine Intelligence,* pp: 1330 - 1334 Vol.22(11), November 2000

[Zoghlami97] I. Zoghlami O. Faugeras R. Deriche. Using geometric corners to build a 2d mosaic from a set of images. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* pp. 420-425, 1997.