TOWARDS GENERAL-PURPOSE IMAGE REGISTRATION

By

Gehua Yang

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: Computer Science

Approved by the
Examining Committee:

Charles V. Stewart, Thesis Adviser

Badrinath Roysam, Member

Daniel Freedman, Member

Richard Radke, Member

Rensselaer Polytechnic Institute
Troy, New York

June 2007
(For Graduation August 2007)

TOWARDS GENERAL-PURPOSE IMAGE REGISTRATION

By

Gehua Yang

An Abstract of a Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: Computer Science

The original of the complete thesis is on file
in the Rensselaer Polytechnic Institute Library

Examining Committee:

Charles V. Stewart, Thesis Adviser

Badrinath Roysam, Member

Daniel Freedman, Member

Richard Radke, Member

Rensselaer Polytechnic Institute
Troy, New York

June 2007
(For Graduation August 2007)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

I would like to express my deepest gratitude to my family — my mother and my two sisters. Their unwavering support and their belief in me have been invaluable to me during my seven-year-long study at RPI across the globe. During my study at the other side of the globe, my eldest sister has given birth to an adorable baby girl, whose innocence as well as naughtiness has brought us fun and happiness. I would like to dedicate this thesis to the memories of my father, who passed away when I was 12 years old. His integrity, his strictness, and his kind love to us all have been such a precious gift to me.

Though away from me across the Atlantic Ocean, my fianceé, Fei, has brought me enormous joy and happiness with her cunning wits. Her encouragements and efforts enabled me to push on in times of distress.

I would like to thank my adviser, Professor Charles (Chuck) Stewart, for his guidance throughout the Ph.D. study, his patience during my fruitless years, and his unshaken belief in my ability. I thank my committee members, Daniel Freedman, Rich Radke, and Badri Roysam, for their support, ideas, and advices on thesis writing.

The presence of my teammates, Ken Fritzsche, Charlene Tsai, Michal Sofka, Matt Turek, Chao Chen, Brad King, Juda Becker, Avi Kelman, Eric Smith, and Alper Ayvaci has made my study so unforgettable. Charlene's sense of humor and infectious excitement, Michal's beer-drinking spirit, and Chao's being-so-much-fun-to-tease have boosted my morale all along.

Last, but not the least, I wish to thank all my friends for lighting up my life during my stay at Troy. In particular, Clarence Chan and I wandered around the streeets of Troy during our first year at RPI and once had beer on the roof top of Amos Eaton. Sean O'Connell and I ran uphills and threw up at the 8th Street right after having pizza. Chunzhi Dong often invited me sincerely to his hot-pot parties that were too spicy for me. Xiaoyun Ji succeeded within 10 minutes in persuading me to make a sky diving. Xi Zhang's presence and the following marriage with

Chao have made the last winter more endurable. All of them have made my time in graduate school enjoyable well beyond mathematics and computer monitor screens.

The work was funded in part by the National Sicence Foundation (NSF), in part by the Gordon Center for Subsurface Sensing and Imaging Systems (CenSSIS), and in part by the US Army Intelligence and Security Command.

ABSTRACT

This thesis presents four image-registration-related methods. The first two methods aid the construction of a fully-automatic retinal image registration system, called RIVERS.

The first method, automatic masking, is a preprocessing technique for image registration that separates the retinal surface from the background in each image. It is especially important when aligning retinal images scanned from slides or images acquired from different cameras.

The second method, a covariance-driven refinement technique, is developed to handle pairs with extreme low overlap (e.g. 5%) in the context of aligning a set of sparsely-overlap images.

The third method, named GDB-ICP, is an automated 2d-image-pair registration algorithm using a hypothesis-and-test strategy and an extension of the Dual-Bootstrap method for refinement. This algorithm is capable of aligning images taken of a wide variety of natural and man-made scenes as well as many medical images, tolerating low overlap, substantial orientation and scale differences, large illumination variations, and physical changes in the scene. An important component of this is the ability to automatically reject pairs that have no overlap or have too many differences to be aligned well. Experimental results on a data set of 22 challenging image pairs show that the algorithm effectively aligns 19 of the 22 pairs and rejects 99.8% of the misalignments that occur when all possible pairs are tried.

An extension of the algorithmic principle of GDB-ICP, the fourth method, a 3d-to-2d alignment algorithm, is applied to estimate the location of a hand-held camera with respect to a 3d model augmented with texture information. Little prior knowledge is assumed about the camera location. A key issue is that initially the model-to-image mapping is well-approximated by a simple 2d-to-2d transformation based on a local model surface approximation. However, the algorithm must transition to the 3d-to-2d projection necessary to solve the position estimation problem. The experiments are conducted on a collection of 9 range scans and 60 image cover-

ing approximately a 100m x 100m region of RPI campus. The algorithm successfully and correctly determines the camera location of 52 images, while indicates it cannot find an alignment for the remaining 8.

CHAPTER 1
Introduction

1.1 Motivation

During the last decade, digital imaging devices have been undergoing rapid development. For instance, the off-the-shelf digital cameras now have a resolution beyond ten million pixels and camcorders have a video resolution about half million pixels, an increase of more than 10-fold compared to ten years ago. The same changes are also happening in the medical area — more and more images are acquired to aid diagnosis, treatment planning and treatment monitoring. Pushed by technological advances in image-capturing devices, the challenges have gradually shifted from image acquisition to image content analysis and information gathering. Images taken at different times, or by different sensors are to be combined, fused, or even analyzed before presenting them to people. Given current image acquisition technology, if images are not presented carefully, one could easily be drowned in a sea of digital data.

Image registration (or image alignment) is a crucial step towards automatic image analysis. It is a process of finding an optimal transformation function (or functions) between two or more overlapping images taken from different view points or taken at different times, such that these functions map pixels of each image onto one common coordinate system. Only after the images have been transformed to the common coordinate system can many applications perform well. These applications include (but are not limited to) change detection [76, 172, 219], mosaic construction [23, 180, 191], image fusion [106, 141, 162, 166], High Dynamic Range (HDR) imaging [48, 230] and super resolution [34, 187].

To register two consecutive video frames is a well-understood task [94, 179, 206, 226]. However, the image registration task becomes significantly harder or even daunting when images are taken from widely separated view points, at different times, with different illumination conditions, or in different modalities. These challenges have been intensively studied in current research, but usually in restricted

environments. For instance, many algorithms have been developed to handle wide baseline stereo (such as [143, 170, 225]), but few of them are sufficiently robust to small overlap, drastic illumination changes, or image modality changes.

To address these issues, we propose a fully-automatic image registration system with the following characteristics:

- **Automatic**: given a set of images and a transformation model, the system is to perform image registration without any human intervention and without parameter tuning.

- **Self-deterministic**: it has to decide which pairs in this set of images have overlap and therefore can be aligned. This is important because any false alignment could adversely affect the overall registration result.

- **Accurate**: the resulting transformations should have alignment error as small as possible, preferably less than a pixel. This is important to many applications, particularly change detection.

- **Efficient**: a naive way to register a set of N images is to align all $N(N-1)/2$ pairs, which requires $O(N^2)$ time complexity. This is too expensive for a large number N.

- **Robust**: the system should be able to handle some combination of low overlap, occlusions, illumination differences (e.g. day and night), shadows, substantial scene changes and even different modalities.

- **General-purpose**: The system should also be able to handle a variety of imaging modalities. For instance, compared to natural scene images, images of human retinas acquired by fundus cameras have little texture, but usually have clear vascular structures. As a result, an algorithm exploiting only texture could perform badly on retinal images, whereas one exploiting only thin-and-long (vascular) structures could perform badly on natural images.

  The system should be able to handle other image dimensions. In an instance of this, a 2d image can be aligned with respect to a 3d world model and therefore the location of the camera can be determined.

The last characteristic, general purpose, is difficult to achieve as there is an unlimited variety of scenes and imaging devices. To focus our efforts, in this thesis we concentrate on two imaging types: natural-scene images and medical images, particularly retinal images, for 2d-to-2d image registration. We choose retinal images because 1) they are different from natural-scene images — little texture and dominant vascular structure — and 2) the registration of retinal images is useful in the diagnosis of retina-related diseases. In addition, applying the same principal as in 2d-to-2d image registration, we will study the registration between an image and a 3d model augmented with texture information.

1.2 Retinal Image Registration

Retinal images are often taken for the diagnosis of a variety of retina-related diseases, such as glaucoma, diabetic retinopathy and macular degeneration. These diseases affect a large and growing percentage of the population. Early detection and treatment, especially of diabetic retinopathy, can significantly improve long-term visual health [54]. To aid the diagnosis of retina-related diseases, image registration techniques can be applied to align a set of retinal images. Here are a few applications for retinal image registration:

- High-resolution, large-field-of-view mosaics formed by multiple retinal images(see Figure 1.1) present a more complete view of a patient's retina surface [31]. Therefore it is easier for doctors to locate pathologies and important structures by looking at the mosaic, and hence aids the diagnosis.

- Screening of diseases such as diabetic retinopathy can involve analysis of retinal images taken from different modalities over many years. Automatic algorithms applied to the already-aligned images to detect temporal changes [172, 178] or pathologies [229, 243] can be used to speed up this screening process and reduce the amount of labor by an ophthalmologist. [43, 211].

- "Spatial Referencing", a real-time image registration technique, locates specific points retinal images acquired in real time by registering this image with

the pre-established spatial image map [116, 190]. This automatic "spatial reference" can greatly aid laser photocoagulation therapy, the only proven treatment for leading blindness-causing conditions, such as diabetic retinopathy and age-related macular degeneration [107, 149, 150, 152].

- Information on aligned multimodal images can be fused together to enable an ophthalmologist to better diagnose disease and judge the outcome of surgery. The detection of microaneurysms is important to the diagnosis of diabetic retinopathy. Although Fluorescein Angiogram (FA) images show most microaneurysms, some are only detected in red-free images [84, 104]. Therefore, the integration of information on FA and red-free images provides a better way to detect microaneurysms (see Figure 1.2).

When constructing mosaics to build complete, non-redundant views of the retina, for a variety of reasons, including cost, patient comfort and patient volume, images of the retina, acquired using a fundus camera, must be taken as quickly as possible and with as few shots as possible [51]. At the same time, it is also advantageous to bring images acquired at different dates, possibly across several years, into one common coordinate system. For instance, images transformed to the common coordinate system can be used to form the basis for change detection. Combining these requirements, it is important to construct a wide-field-of-view mosaic from a sparse set of retinal images and possibly from different acquisition dates.

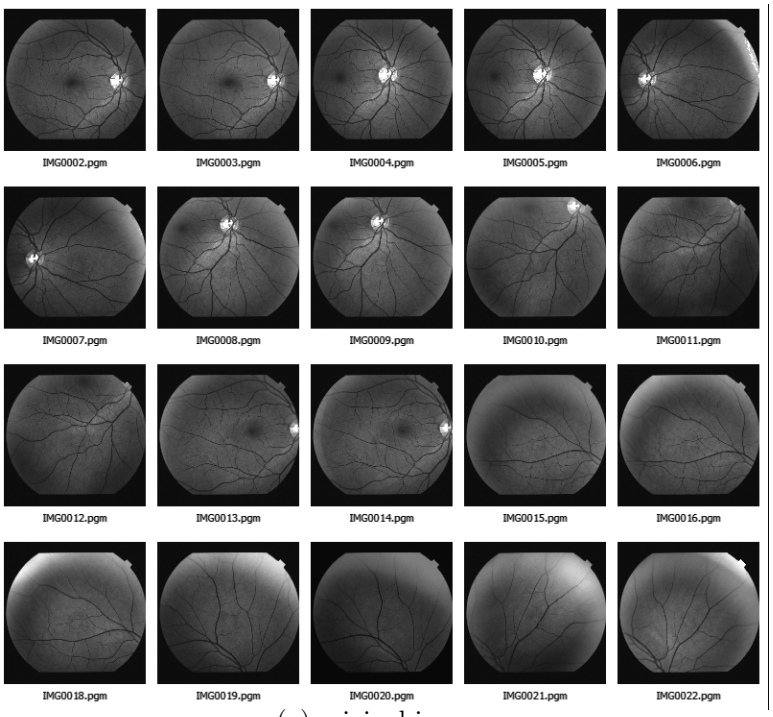
We present a full-automatic retinal image registration system capable of aligning images with small overlap (as small as 2% in multi-image registration) and robust to longitudinal changes in the eye.

1.2.1 Overview of RIVERS

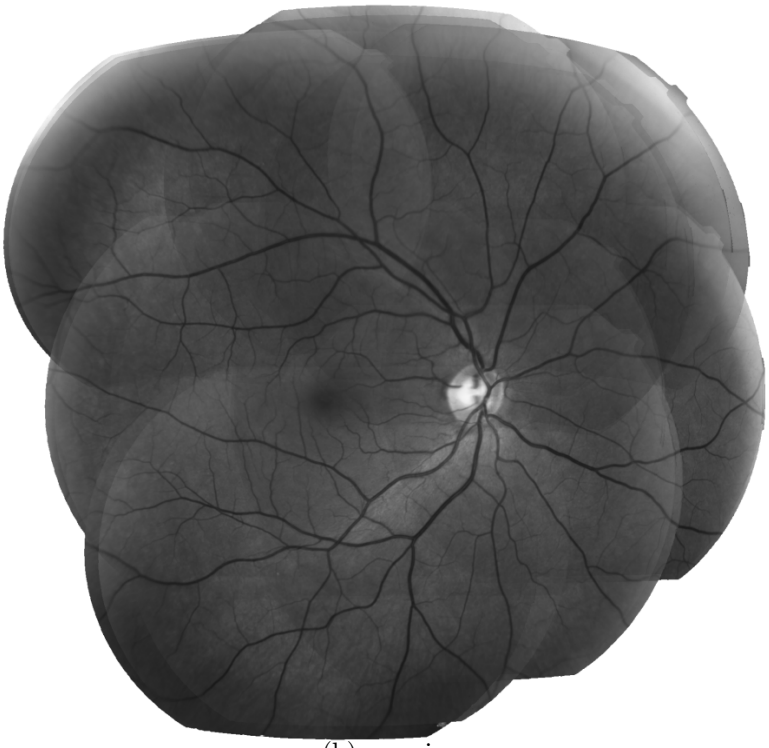
Retinal Image Vessel Extraction and Registration System (RIVERS) is a fully automatic system for vasculature detection and alignment of retinal images[1].

This complex system was developed through the collective efforts of a number of students and faculty. Starting around 1999, Can presented an exploratory vessel extraction[29], landmark and trace based pairwise registration[32], and linear

[1]online access is available through http://vision.cs.rpi.edu/RIVERS

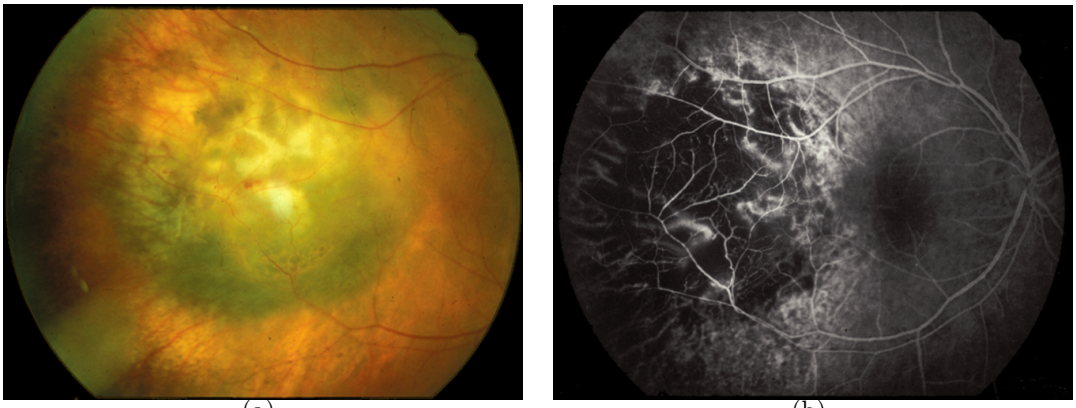


(a) original images

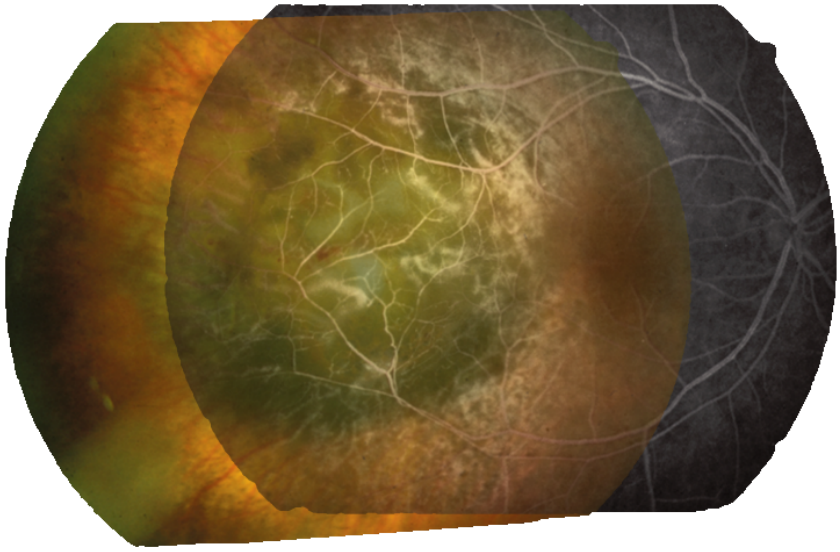


(b) mosaic

Figure 1.1: (a) shows a set of retinal images and (b) shows the mosaic of these images after registration.



(a) (b)



(c)

Figure 1.2: Multimodal registration of (a) color fundus image and (b) one frame of fluorescein angiogram (FA) images of an eye with melanoma. The bottom shows the mosaic.

estimation joint registration[31]. Later on, Tsai took over the pairwise registration and introduced the Dual-Boostrap ICP algorithm with Stewart[199] around 2003. The author has been working on the RIVERS since 2001 and contributed two components: masking and covariance-driven refinement, the details of which will be discussed in Section 1.2.2 and Section 1.2.3, respectively.

RIVERS has six components as shown in Figure 1.3. Combining these techniques , it has achieved great success. In a validation study by Tsai [223], on a set

1. **Masking:** A binary mask image is needed to remove non-relevant backgrounds in a retinal image (see Figure 1.4). The background is produced by an internal mask inside the fundus camera and sometimes contains identifying tags. The shape of the mask differs between images taken by different cameras, images scanned from slides, or images with different degrees of field-of-view.

2. **Vessel Extraction:** Centerline points of vessels are extracted using a exploratory method, which relies on recursive tracing of the vasculature based on a localized model. This work was first introduced by A. Can in [29], and then improved by K. Fritzsche [63]. These vessel centerline points may then be used to drive the image registration process.

3. **Pairwise Registration:** The goal is to determine the alignment between a pair of given images. Feature-based algorithms were developed by Can *et al.* based on vessel crossovers and bifurcations [30], and this was then extended to use vessel centerline points [32]. Later on, Stewart and Tsai took a step further and developed the Dual-Bootstrap ICP(DB-ICP) algorithm [199]. The ability to "bootstrap" the reliable and accurate alignment from a minimal number of landmark correspondences makes it possible to register images that are badly affected by disease and images with low overlap.

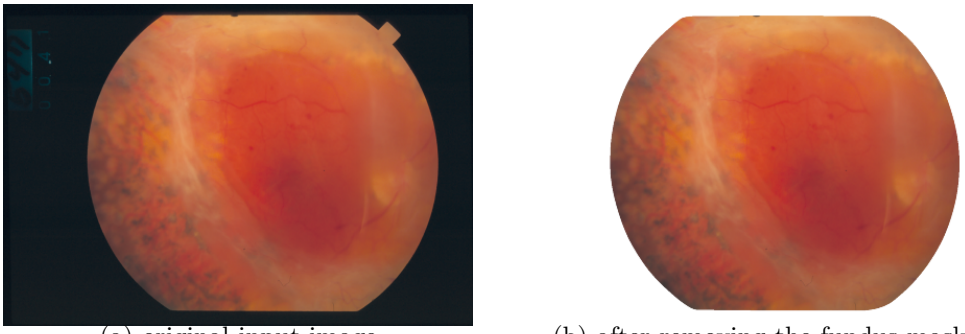
4. **Joint Registration:** Taking the results from the pairwise registration, this step jointly estimates the transformation parameters for all the images. This algorithm was first presented by A. Can in [31]. It linearly and simultaneously estimates all transformations by incorporating direct constraints (from non-anchor image to the anchor[a]) and indirect constraints (between non-anchor images), and estimating transformations parameters that maps each of the non-anchor images to the anchor space.

5. **Covariance-Driven Refinement:** Even though joint registration includes constraints from all registered pairs, misalignment can still occur between the images that have (usually small) overlap but fail to register. These images pairs may be aligned in the context of joint image registration, using the covariance matrix of the transformation estimates to guide the search for new correspondences.

6. **Mosaic Formation:** Given the transformations produced by the joint registration, a mosaic image is generated by mapping pixel intensities into the anchor coordinate system [31]. More sophisticated blending methods, described in [207], can be used to construct seamless mosaics.

[a] An anchor is an image chosen to position the rest of the mosaic upon. It can be any of the images

Figure 1.3: Diagram of RIVERS



(a) original input image (b) after removing the fundus mask

Figure 1.4: The masking of one retinal image: (a) is the input image and (b) is the output after segmenting out the portion of the fundus camera mask

of 855 retrospective retinal images taken from 18 eyes with four common diseases: Diabetic Retinopathy, Vein Occlusion, dry and wet Aged-related Macular Degeneration, this system successfully aligned 853 images and built 18 mosaics, with one mosaic for each eye. The DB-ICP algorithm successfully aligned 99.5% of the image pairs having a sufficient set of common features and 78.5% overall. The system also routinely handles normal retinal images.

Within this six components of RIVERS, the author developed two components: the automatic masking and the covariance-driven refinement. These will be discussed in detail in the following two subsections.

1.2.2 Automatic Masking

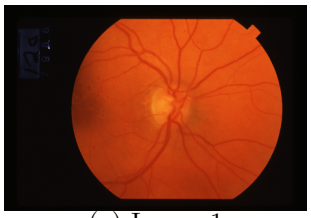
Masking is a process of locating the projection of the retina in a fundus camera image. The problem arises in retinal image registration since fundus cameras produce images in which the retina surface is only captured in a mostly-convex, curved region centered in the image (Figure 1.4); the rest of the image is often dark and homogeneous, due to a mask inside the camera blocking light going through the camera. Our goal is to develop an algorithm that automatically segments the retina region, producing a binary *mask image*, where 1 represents (foreground) retinal regions, and 0 represents the (background) mask.

The mask in retinal images interferes with the image analysis algorithms and

produces unfavorable effects in both the computation aspect and the visualization aspect. Computationally, automatic image analysis algorithms, including vessel extraction [11, 29, 63], optic disk detection [89], and change detection [153, 172], often compute statistics based on the image content. The presence of the mask area in the images affects accurate computation of the statistics and leads to incorrect results. For instance, [63] gathers local statistics on contrast and brightness levels to detect seed location on blood vessels. If the dark region of mask is included in the computation, the statistics are offset from the true values, resulting in both missing seeds and incorrect seed locations. Also, during retinal image registration [31, 32, 199, 222], the strong edge response between the retina surface and the mask areas may pull the alignment in the wrong direction as it favors aligning the outlines of the masks in the two images. Visually, during mosaic construction, pixels in the mask area in one image blended with pixels on the retina surface in another image severely distort the final mosaic (Figure 1.5c). In comparison, the mosaic constructed with the mask images is much cleaner (Figure 1.5d).

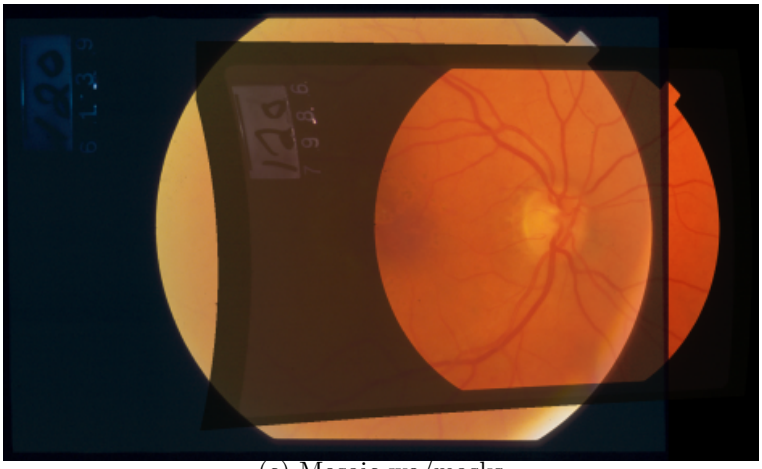
At a first glance, masking appears to be a simple problem. When working with a fixed field of view, using a single, digital fundus camera, the mask is the same for all images. Hence, a single mask image may be manually-specified and used for all subsequent images. However, when working with images acquired from different cameras, or when working with scanned slides originally taken from a traditional film camera (especially for retrospective studies), the corresponding mask image differs from one image to another. To manually segment the mask from each individual image is tedious and out of the question when conducting a large scale retrospective study. Thus, an automated masking technique is needed to carry out the task.

In the choices of automatic masking techniques, one's thought may immediately cast upon simple techniques, such as a straightforward application of thresholding, perhaps with morphological operations. For instance, [65] applied global thresholding, followed by median filter on the intensities. After examining a broad selection of fundus retinal images, however, we believe that such simple techniques do not suffice. Example images are shown in Figure 1.6 to indicate the challenges in the automatic masking: (a) the digitization process of films/slides produces double

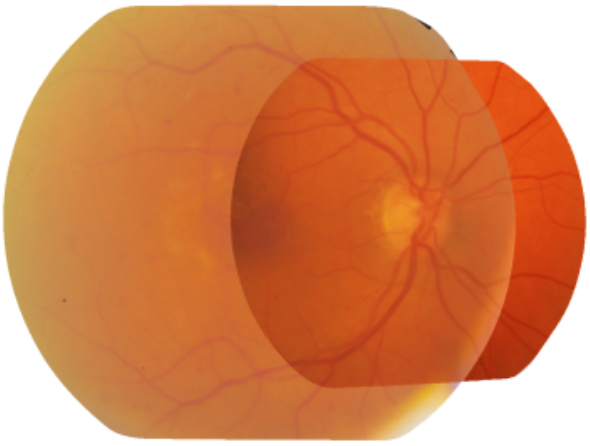


(a) Image 1

(b) Image 2



(c) Mosaic wo/masks



(d) Mosaic w/ masks

Figure 1.5: The importance of masking: A mosaic is produced after aligning Image 1 and Image 2. If no masks are provided, the dark mask area is blended into the mosaic (c). After applying the appropriate mask for each image, the mosaic is much cleaner.

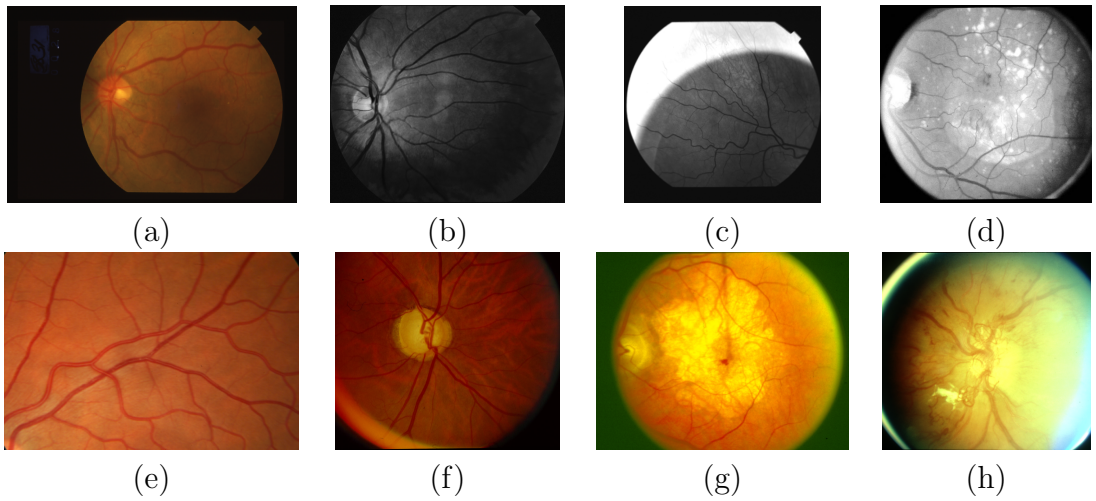


Figure 1.6: Examples of a variety of retinal images, which demonstrates various shapes and color of mask region. (a) and (b) are further discussed in Figure 1.7

boundaries in the images, one from the fundus mask, and one from the boundary of the slide frame (shown in Figure 1.7 after contrast enhancement). In addition, some slides have numerical identifier tags in the mask region, which are brighter even than the retina surface. Next, illumination of retinal images is uneven, due to an optical aberration called vignetting [89]. The lens of the camera works in conjunction with the lens of the eye to form the image by capturing the light reflected off the retina surface and through the pupil. As the position of an eye relative to the camera varies from image to image, the exact properties of the vignetting also vary from image to image. This vignetting results in a combinations of dark regions (b and h), bright regions (c), and halos (h). In particular, the dark region of retina on (b) is hardly distinguishable from the mask and can only be seen after manual contrast enhancement (shown in Figure 1.7). Other challenges include (g) noise and color artifacts in the mask area ; (h) pathologies and the effects of disease that change the reflectance properties of the retina. Finally, in some images, particularly the ones digitized from slides, the mask boundary may have different shapes from the expected ones (e)-(h). In Chapter 3, we are going to develop an automatic and robust masking method to address these difficulties and challenges.

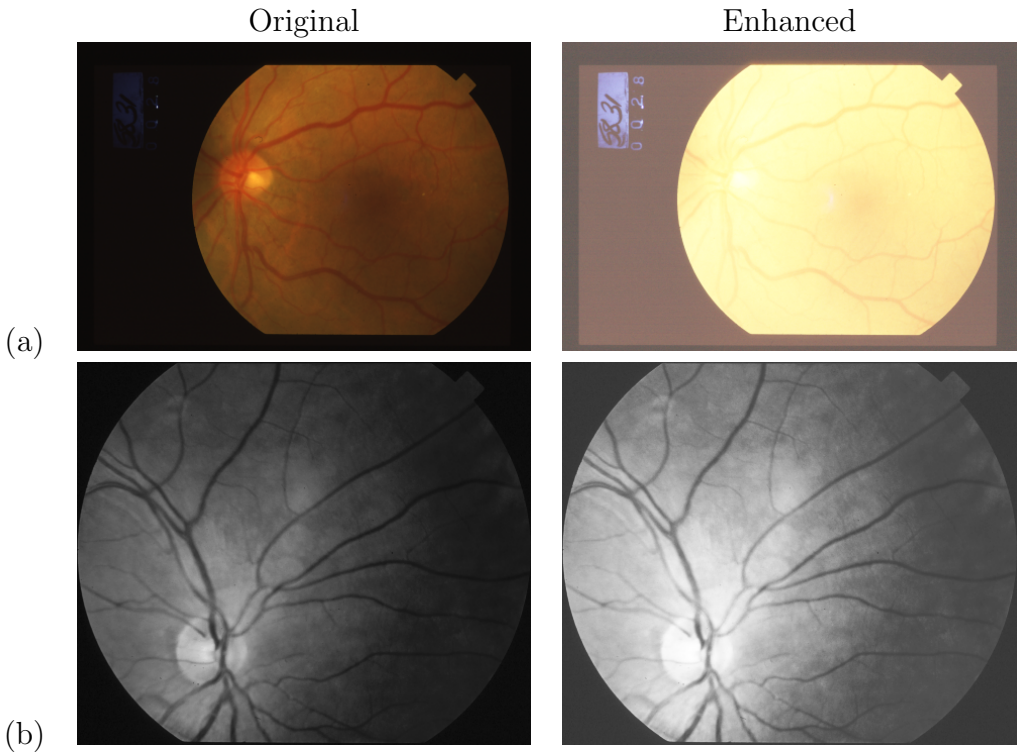


Figure 1.7: Contrast and brightness enhancement to show the difficulty in masking. The original image is to the left and the enhanced one is to the right. The 1st image is taken from Figure 1.6a and is enhanced to show the double boundaries in the scanned slide. The 2nd image is taken from Figure 1.6b and is enhanced to show the hardly-distinguishable mask boundary on the bottom right due to vignetting.

1.2.3 Covariance-driven Refinement

In RIVERS, after removing the fundus mask, the retinal images are aligned using the extracted vessel centerline points. After the joint registration as shown in Step 4 in Figure 1.3, a refinement step is required to align image pairs that fail the pairwise registration due to small overlap and insufficient constraints.

As we mentioned before, it is important to construct a wide-field-of-view mosaic from a sparse set of retinal images [51]. However, when the image set is sparse, some image pairs may have small inter-image overlap, causing the pairwise registration algorithm to fail and misalignments to show in the mosaic (Figure 1.8). It is difficult to improve the pairwise registration algorithm to align these image pairs because there is not enough common information available in the region of the over-

lap to produce a stable transformation. We propose to solve this problem in the context of multi-image registration.

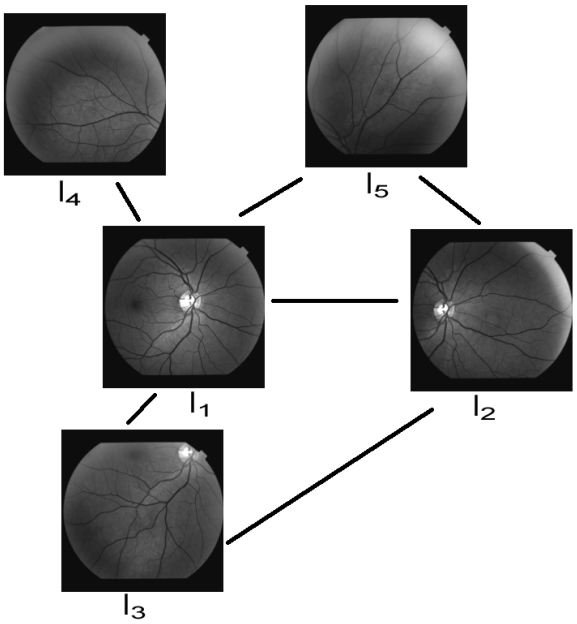
The multi-image registration problem has received a significant amount of attention both in the computer vision literature [160, 180, 191] and the ophthalmology literature [49, 231]. Some techniques focuse on mosaicing video sequences [180, 191], where the overlap between images tends to be relatively high. Some techniques focus on wide-baseline matching for estimation of the fundamental matrix [181]. Some techniques have addressed the question of low image overlap, using the global inferencing of topological relationship between images to generate additional inter-image constraints [179]. These techniques focus on obtaining a pairwise transformation estimate for the pair with small overlap. However, when the overlap is extremely small, e.g. 5%, there is such little information that any pairwise registration technique is unlikely to succeed.

We are going to address this problem in the context of multi-image registration. The details of the algorithm is presented in Chapter 4.

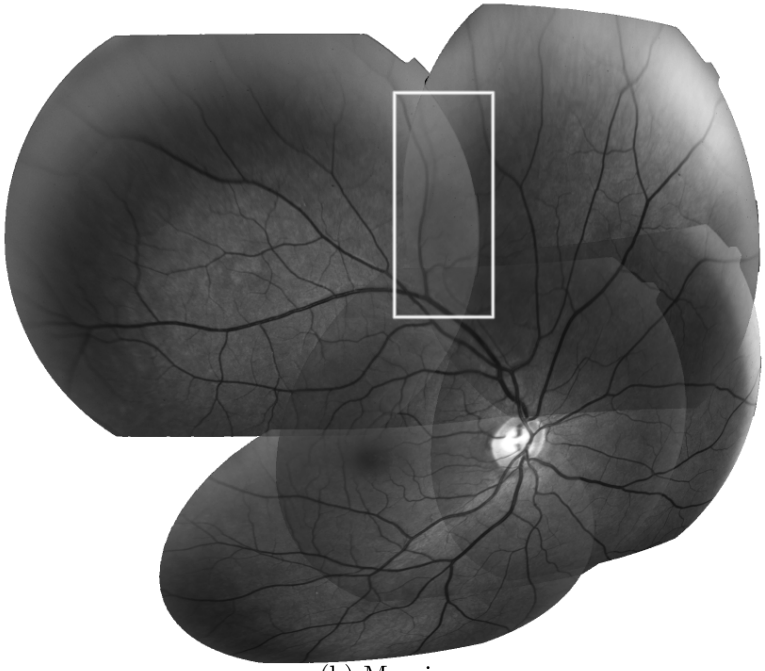
1.3 Generalization

While extremely effective, the registration algorithm of RIVERS is restricted to 2d retinal images. We would like to extend the success to handle other imaging types as well. Preferably, the new system should be able to automatically determine image relationships (including aligning images) given a set of images, which may be taken from different view points, at different times, or with different modalities. We would also like to determine image/scene relationships in different dimensions (e.g. 3d versus 2d).

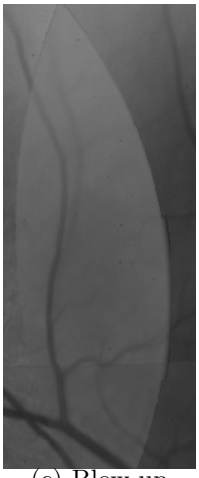
Meanwhile, as the new system works on various imaging types, we would still like to push it further to improve retinal image registration. In particular, RIVERS may fail when the images are blurry, in low contrast, covered by pathologies, or affected by diseases like geographic atrophy, because either vessels are too difficult to extract or not enough vessels appear in the images. One solution is to improve vessel segmentation methods to work on low contrast images and on thin vessels, such as the one proposed by Sofka and Stewart [193]. But our goal is a generic



(a) Connecting graph



(b) Mosaic



(c) Blow-up

Figure 1.8: Mosaic construction of retinal images. (a) shows a graph of retinal images as vertices and edges between vertices if the associated images are aligned via pairwise registration. Notice that image 4 and image 5 overlap (initial overlap 5.8%) but are not adjacent in the graph. This leads to inaccuracies in the mosaic (b), highlighted by the rectangular area. Blow-up of the rectangular region is shown in (c).

method for image registration, not specific to retinal images.

The focus of the generalization is cast upon the feature extraction method, the initialization, and the refinement technique for pairwise image registration. Looking back at the diagram of RIVERS system (Figure 1.3), the multi-image registration and the following covariance-driven mosaicing require only feature correspondences from image pairs and hence can be applied as a general approach. Masking is a procedure specific only to retinal images. The remaining two components — feature extraction and pairwise registration — have great impact on the overall system performance as it depicts which genre of image pairs can be aligned. Thus, we concentrate on these two components in the following discussion, introducing generalizations for handling other imaging types (Chapter 5) and different dimensions (Chapter 6). After the success of this generalized image registration algorithm on image pairs, it is straight forward to apply the subsequent multi-image registration and covariance-driven mosaicing.

1.3.1 Pairwise 2d image registration

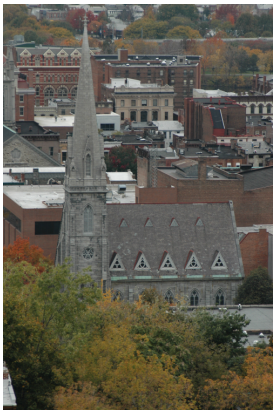
In general settings, images can be taken by different imaging modalities, in different illumination conditions, with different rotation, scale and translation, or in the presence of physical changes. Some images are blurry while some are sharp in focus. Some are full of texture while some have little. Some are taken in color while others in grayscale. Much of this variety is captured in a test suite of 22 image pairs[2] we have gathered, some of which are shown in Figure 1.9, Figure 1.10, Figure 1.11, and Figure 1.12. The suite includes image pairs taken of indoor and outdoor scenes, in natural and man-made environments, at different times of day, during different seasons of the year, and using different imaging modalities. It includes image pairs with low overlap (e.g. 2%), substantial differences in orientation (90 degrees), and large changes in scale (up to a factor of 6.4).

A general-purpose registration algorithm should be able to align each of these image pairs with high accuracy without prior knowledge on the image pair. Moreover, such an algorithm should be able to indicate that two images *can not* be aligned

[2]Both the test suite and our software are available at http://www.vision.cs.rpi.edu/gdbicp/


White Tower


Dashpoint (2% overlap)


Downtown winter day, summer day, summer night, respectively.

Figure 1.9: Some images from our 22-image pair dataset. The three "Downtown" images produce three of our test pairs.

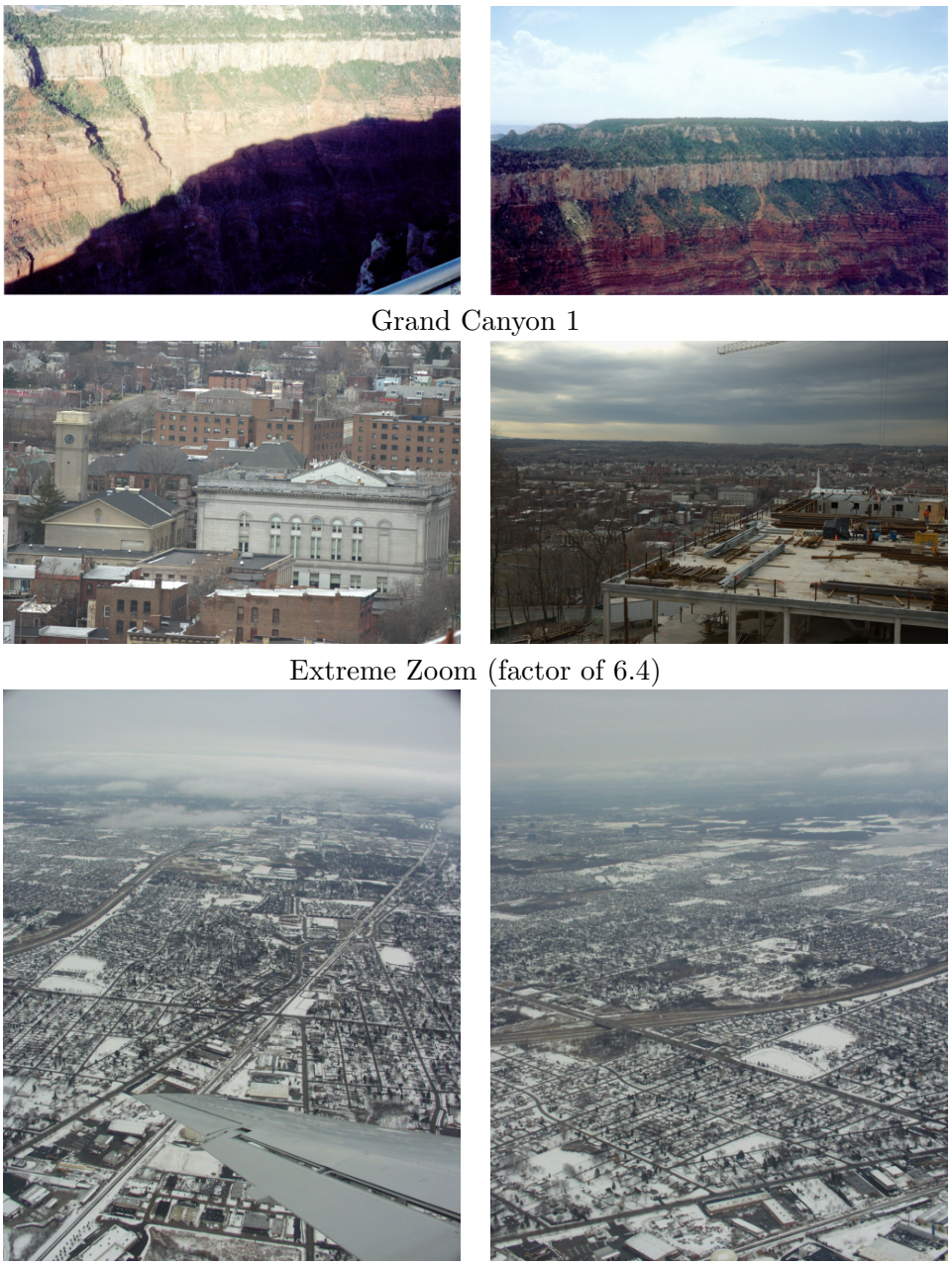
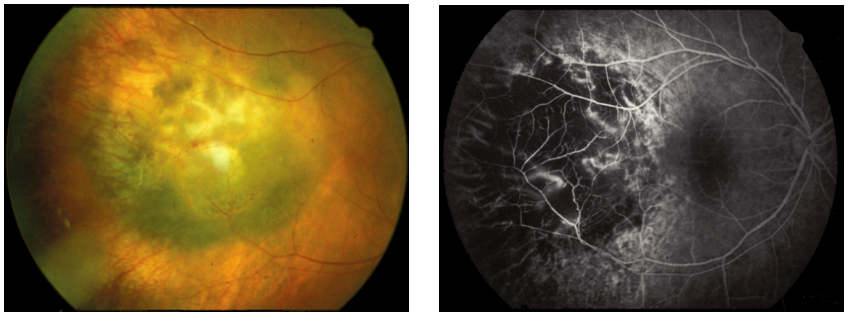


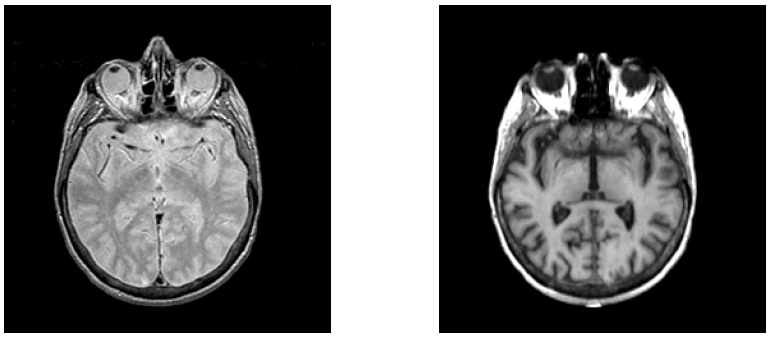
Figure 1.10: Some images from our 22-image pair dataset.

Brugge Tower

Boston Library



Melanoma



Brain PD-T1

Figure 1.11: Some images from our 22-image pair dataset.

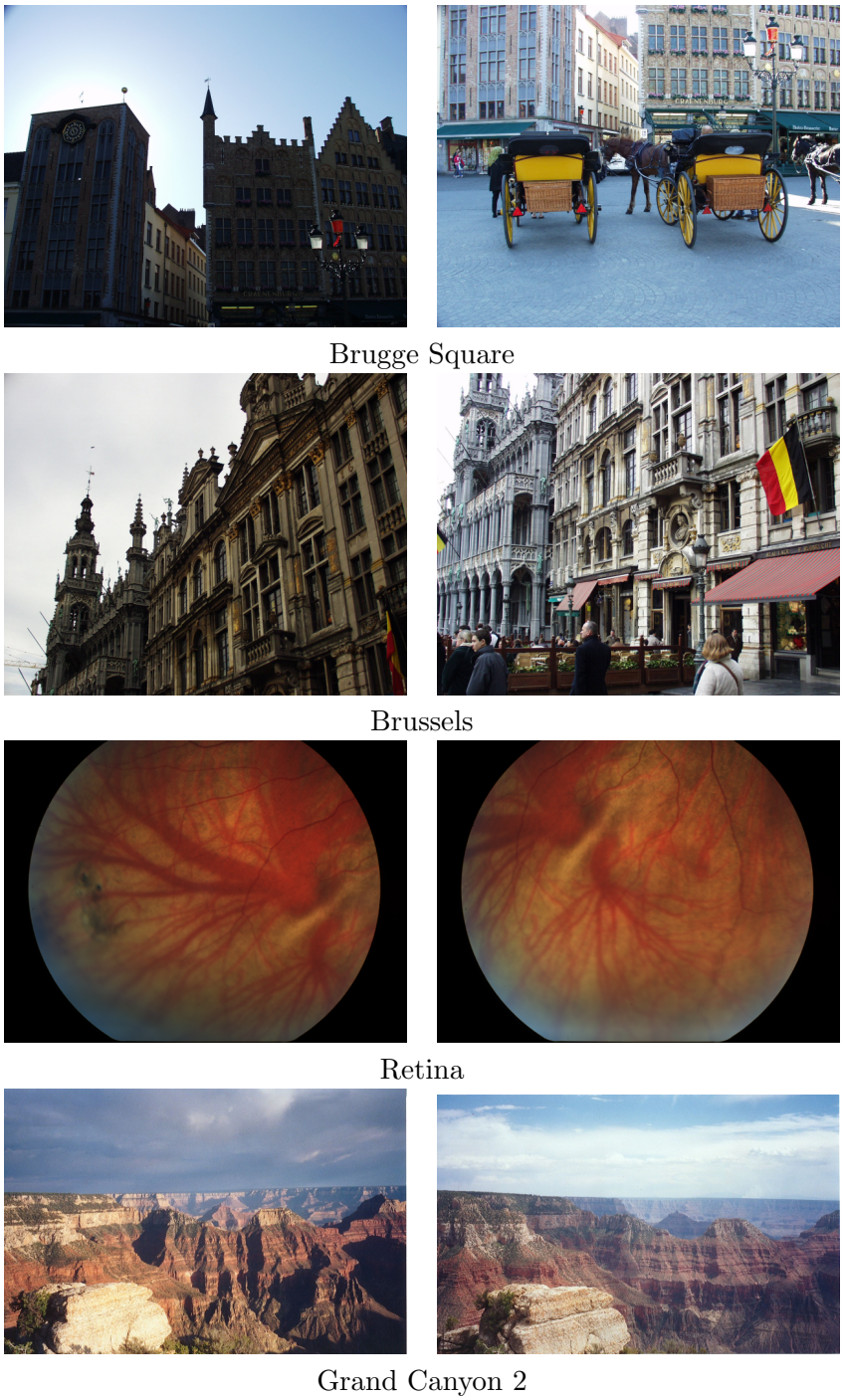


Figure 1.12: Some images from our 22-image pair dataset.

either when the images truly do not overlap or when there is insufficient information to determine an accurate, reliable transformation between images. Such a registration algorithm will have numerous applications ranging from mosaic construction [1, 23, 28, 205, 207, 228] to change detection and visualization [172].

Three primary technical challenges must be addressed in order to solve this problem: initialization, estimation, and decision.

- While automatic initialization is not a significant problem for aligning images in a video sequence or for multimodal registration of images taken from roughly pre-aligned sensors, it is a major concern for more general-purpose registration.

- In combination with initialization, the estimation process must tolerate position, scale, orientation and illumination differences. Moreover, estimation must accommodate the possibility that there is no relationship between the intensities for a large fraction of the pixels in the two images. For example, in the Summer-Winter pair from Figure 1.9, snow on the roofs in winter produces homogeneous intensity regions, whereas these roofs appear as dark, textured regions in the summer image (Figure 5.3). Because of this, an effective estimation technique should automatically and adaptively exploit what is consistent between the images.

- Decision criteria are required not only to choose among different estimates obtained from different starting conditions, but also to decide when the images may not be aligned at all. The criteria play an crucial role in a panorama construction system [23], where pairwise alignments must be determined automatically from a set of unordered images, and in a object recognition system [232], where the object pose estimate is used to determine the presence of the object. The need for effective decision criteria is particularly acute when handling low overlap and large changes in orientation, illumination and scale due to the extremely large search space of initial estimates.

  registration, vasculature usually has enough complexity that a measure of alignment error is sufficient in determining the success on registering a pair of images. In other situations such as natural-scene images, however, due to

the texture and locally similar structures, this simple decision criterion often results in mis-registration. More sophisticated decision criteria are needed to cope with the variability between images.

We propose an fully automatic image registration system — the Generalized Dual-Bootstrap ICP algorithm — to address all three challenges. This algorithm uses a hypothesize-and-test strategy and an extension of the Dual-Bootstrap ICP algorithm [199], originally designed for the RIVERS system. Each local initial transformation is treated as one hypothesis. By exploiting the assumption that a large fraction of the scene is rigid and therefore a single image-to-image transformation function is appropriate, each initial lower-order local transformation is "expanded" into a global higher-order transformation using the Dual-Bootstrap method as the refinement procedure. After the region expansion and the refinement of the transformation, the resulting global transformation is checked with the decision criteria, testing for accuracy, stability, and consistency. The transformation that fails the decision criteria is discarded and the next hypothesis — a new local initial transformation — will be tested using the same approach, until one transformation passes the decision criteria. This is repeated until a certain number of the hypotheses have been tried. The details of the algorithm will be illustrated in Chapter 5.

With the proposed algorithm, we are able to align a variety of 2d images with appearance changes or in different modalities. We would like to apply the same algorithm principle to other image registration problems. The problem instance explored in this thesis is determination of the location of a hand-held camera with respect to a 3d world model.

1.3.2 Camera Location Estimation

We address the problem of finding the location of a hand-held camera with respect to a 3D world model augmented with photometric texture information. Applications include automatic navigation [95, 155], automatic integration of new information into a modeling system [195], and automatic generation of model-to-image overlays [242]. All of these will become increasingly important as modeling systems, such as Google Earth, progress toward more accurate 3d representations. For the

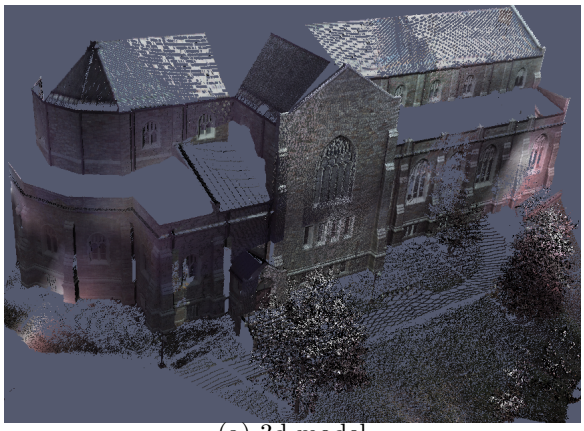
experiments in this thesis, the position of the hand-held camera is known within 100 meters range of its true position for street-level images, but nothing is known about its orientation or intrinsic parameters.

Inferring the camera location requires both establishing correspondences between the model and the test image and estimating the model-to-image camera projection, effectively calibrating the hand-held camera [224, 241]. Several complications in the data, some illustrated in Figure 1.13, make this problem challenging, including (a) a large search space of camera poses, (b) occlusions, (c) differences in viewpoint and illumination between the test image and the images acquired by the pre-calibrated camera attached to the range scanner, (d) buildings and other objects with repetitive appearance, and (e) physical changes in the scene between model construction and test image acquisition.

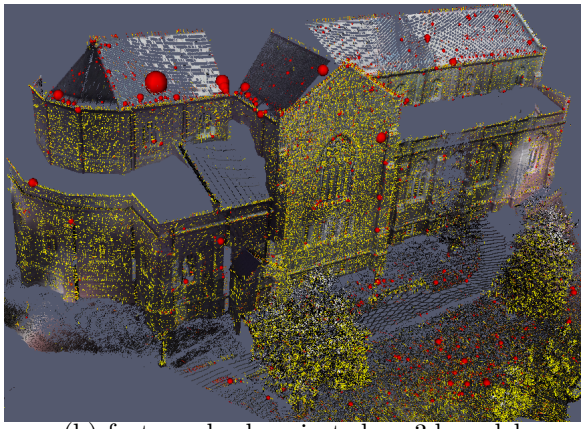
Using the same approach as the 2d image registration algorithm proposed in Chapter 5, we construct an automatic system to estimate the camera location while tolerating appearance changes, physical changes, or occlusions. The new system also uses the hypothesis-and-test strategy and employs the Dual-Bootstrap refinement technique. Key differences are in the hypothesis generation and the camera projection estimation. When considering the hypothesis generation, it is non-trivial to generate a hypothesis using a camera projection model. Hence, in this work, we propose instead to generate a local initial *surface-to-image* mapping using the similarity transformation. Such a hypothesis generation alleviates the difficulty for obtaining an initialization. Meanwhile, it posts a challenge on the camera estimation because, at some point, the aglorithm must make a transition from a surface-to-image mapping to a model-to-image projection. The details of the algorithm, especially the key issue of how and when we make the transition, are described in Chapter 6.

1.4 Organization and Summary of Contributions

Chapter 2 provides a literature overview on image registration methods. Chapter 3 presents the masking technique for retinal images. Chapter 4 presents the covariance-driven refinement technique for registering of a set of sparsely-overlap images. These above two chapters focus on the application of automated retinal

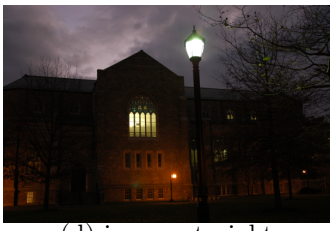


(a) 3d model



(b) features backprojected on 3d model

(c) zoom-in on features



(d) image at night

(e) image after snow

Figure 1.13: 3d Model, features and example test images. (a) shows part of the 3d model, while (b) shows the model features superimposed on the 3d model, with (c) showing a zoomed-in view. In (b) and (c), the back-projected model corners points are represented by spheres whose radius is proportional to the corner point scale, while edge-like features are represented by arrows. (d) and (e) show different test images that our algorithm can accurately localize, even though the images were taken at night and during the winter.

image analysis. These following two chatpers are the emphasis of this thesis: Chapter 5 presents a fully-automatic pairwise image registration system that works on a variety of 2d images. Extending from the algorithm principle proposed in Chapter 5, Chapter 6 presents an algorithm for determining the location of a hand-held camera with respect to a 3d model. At last, Chapter 7 presents the overall discussions, conclusions, and future work.

The four primary contributions are summarized as follows:

- **Automatic Masking (Chapter 3)** We present a fully-automatic segmentation method that removes the fundus mask area from retinal images taken from different cameras or scanned from slides. On a test suite of 2108 retinal images, the algorithm successfully extracted the mask region of 2096 images and failed on 12 images (99.4% success rate).

- **Covariance-Driven Refinement (Chapter 4)** In the context of multi-image registration, we present a refinement method that uses the solution of bundle adjustment as initialization and iterates between the search for new correspondences guided by the covariance parameter estimate matrix and the refinement of the joint solution. This method effectively aligns image pairs with extreme low overlap.

- **Generalized Dual-Bootstrap Image Registration (Chapter 5)**
  We present a fully automatic Generalized Dual-Bootstrap ICP (GDB-ICP) image registration algorithm designed to handle a wide variety of image pairs, including those showing scale changes, orientation differences, low overlap, illumination differences, physical changes and different modalities.

- **Estimation of the Camera Location (Chapter 6)** We present a fully automatic algorithm to determine the location of a hand-held camera with respect to a 3d model. Taking the Generalized Dual-Bootstrap approach, the algorithm is robust to appearance changes. The major contribution here is the automatic transition from 2d-to-2d mapping to 3d-to-2d projection and the overall capability to start from one single keypoint match.

CHAPTER 2
Background

Image registration has long been the subject of research. Publications date back to the early 70's [9, 146]. A large number of papers have been published since then. To get a picture of the scope, the comprehensive surveys on image registration methods published in 1992 by Brown [22] and in 2003 by Zitovâ [244] are good references. In medical domain, Maintz and Viergever presented a survey [129] on medical image registration methods. The intent of this chapter is to introduce the most recent progress related to image registration methods, along with the basics of the problem to cover the work described in this thesis.

There are a number of criteria to categorize and to classify image registration methods. Some are listed in Figure 2.1. The rest of the chapter will focus on these criteria and discuss various methods in the literature based on these criteria.

2.1 Metric to be optimized

Image registration tasks are usually carried out by setting up the search as an optimization problem. The desired transformation function is obtained by solving this optimization problem. The metric used in the objective function of the optimization is a key aspect of image registration method. Three categories will be discussed in this section: feature-based, intensity-based, and mutual information methods. In a way, mutual information is a measure that also depends on intensity values. However, its formulation is very different from intensity-based methods and it has been widely adopted in a number of applications. Hence, mutual information is worth separate attention.

2.1.1 Feature based

Feature-based methods start by extracting image geometric primitives, namely features, such as corners, curves, surfaces, or cylindrical structures [7, 17, 32, 36, 39]. During the image registration, these extracted features in one image are matched

- **Method** - Metric to be optimized
  - Feature/keypoint based
  - Intensity based
  - Mutual information based
- **Transformation functions**
  - global parametrized transformations
  - local/deformable parametrized transformations
  - non-parametrized/optical flow
- **Application**
  - monomodality (multi-viewpoint)
  - multimodality
  - temporal
- **Spatial Dimension**
  - 2d to 2d
  - 3d to 3d
  - 3d to 2d
- **Number of images**
  - two images
  - multiple images (more than two)

Figure 2.1: Classification scheme of image registration

to the ones in another image, either by their appearance similarity or by geometric closeness. During the matching, correspondences are formed between features in the two images. The transformation is estimated from an objective function based on a geometric distance measure.

Let one image be I_p and a second one be I_q. The set of features in I_p is $\mathcal{G}_p$ and features are $\mathbf{p} \in \mathcal{G}_p$, whereas features $\mathbf{q}$ are in the feature set $\mathcal{G}_q$ on image I_q. (Abusing the notation, $\mathbf{p}$ represents both the feature and its location.) The correspondences $(\mathbf{p}, \mathbf{q})$ form a set $\mathcal{C}$. The transformation function from I_p to I_q is $\mathbf{T}(\mathbf{x}; \boldsymbol{\theta})$, with parameter vector $\boldsymbol{\theta}$. (For now we focus only on parameterized transformation functions.) The geometric distance measure between two features

$\mathbf{p}$ and $\mathbf{q}$ is denoted as $d(\mathbf{p}, \mathbf{q})$. Given a correspondence set $\mathcal{C}$, the least-squares objective function for feature-based methods can be written as

$$\min_{\boldsymbol{\theta}} \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{C}} d(\mathbf{p}, \mathbf{T}(\mathbf{q}; \boldsymbol{\theta}))^2. \tag{2.1}$$

In most cases, however, because the correspondence set $\mathcal{C}$ is either unknown or contaminated by a large fraction of outliers, the estimation of Equation (2.1) cannot be carried out by conventional optimization methods. Instead, alternative ways of matching and estimation have been developed to handle this issue.

2.1.1.1 ICP

Our first class of feature-based methods contains the Iterative Closest Point (ICP) algorithms [17, 36, 39] and its variations. Starting from an initial estimate, ICP iteratively (a) maps points (features) from one image, I_p, to the other image, I_q, (b) finds the closest point on image I_q for each mapped point, and (c) re-estimates the transformation using these temporary correspondences in the objective function (2.1). In other words, the correspondence set $\mathcal{C}$ is obtained from forming correspondences with the closest points. The convergence proof of ICP has been shown by Besl and McKay [17] using a Euclidean distance metric for d in (2.1)[3]. In the registration of range data, sensor data points (together with estimated normals) are the features commonly used [16, 19, 39, 64].

In the alignment of intensity images, feature extraction techniques are applied to obtain features to be matched in ICP [32, 75]. Extraction of Harris corners is commonly seen in the literature [80, 184, 245]. Can *et al.* extracted vessel centerline points (edge-like features) for the registration of retinal images [32].

There are two commonly-known problems with ICP: (1) ICP has a narrow domain of convergence and therefore must be initialized relatively accurately; and (2) feature extraction can be unreliable and overly sensitive to the choice of parameters and the image content.

The distance measures used in Equation (2.1) are usually associated with the

[3]This proof does not apply to other metric.

types of features being extracted. For corners or points, a natural distance measure is Euclidean distance [17, 140, 240]. However, the situation is different for points sampled on a curve on a 2d image. As they are discrete samples from a continuous curve, we are not concerned with how far a point moves *along* the curve, but how far a point is *off* the curve. Hence, we want to compute point-to-curve distances. One efficient way to achieve this is to "linearize" the curve — approximate the curve locally with a straight line — in a small neighborhood around the sample point. The point-to-curve distance is then measured by the distance along the normal direction of the straight line [32]. The same linearization is also applied to points sampled on a surface in 3d image [39, 201]. Such a linearization can be viewed as the first-order approximation to the true distance function. Recently, Pottmann and Hofer proposed a second-order approximation that takes local curvature into account [168] by fitting a circle locally tangent to the curve.

2.1.1.2 EM-ICP

Several papers [42, 73, 125] have proposed Expectation Maximization (EM) algorithms [52, Ch. 3] for image registration, where multiple correspondences per feature are simultaneously considered. In this approach, correspondences are considered as "soft" — each associated with a probability of this correspondence being correct. The EM algorithms are carried out by alternating between the computation of this correspondence probability (expectation) and the estimation of transformation parameters (maximization). In [40, 42] multiple correspondences are modeled with a Gaussian mixture model with an isotropic covariance. Later on, Chui and Rangarajan exploited Expectation-Maximization along with simulated annealing and a doubly-stochastic normalization over the correspondence matrix to solve for free deformations [41]. This approach improves the robustness and reliability of ICP.

2.1.1.3 Keypoint Extraction and Matching With Invariant Descriptors

In contrast to ICP where matching is based on geometric distance, keypoint based methods apply matching in appearance similarity space. The term "keypoint" refers to a combination of an interest point (or region) and an invariant descriptor which describes the intensity characteristics in a small neighborhood around the

interest point. The latter part, the descriptor, is the key difference between the definitions of keypoint and feature. Specifically, a descriptor is designed to be invariant to some lower-order geometric transformation, e.g., similarity or affine, and usually additionally invariant to linear intensity changes.

Keypoint methods have received growing attention because of their demonstrated ability to tolerate low image overlap and image scale changes [23, 53, 143]. These methods start with keypoint detection and localization followed by computation of the descriptor that summarizes the image around the keypoint. Existing keypoint extraction algorithms are based on approaches ranging from the Laplacian-of-Gaussian operator [23], information theory [98], Harris corners [143], and intensity region stability measures [134]. A detailed comparison can be found in [145]. Region descriptors proposed are based on steerable filters [61], moment invariants [69], shape context [13], image gradients [122] and Haar wavelets [25]. A comparison of descriptors can be found in [144].

As the descriptors are invariant to geometric transformation, they can be compared directly by computing a similarity score between them. The keypoint matching between two images is carried out by computing similarity scores for all possible keypoint pairs and selecting the most similar ones to form the initial set of putative matches. This brute-force matching requires $O(N^2)$ time complexity for N keypoints. Additional data structures can be used to speed up this matching process. In [23], a k-d tree is used to store keypoints in one image and the matching is carried out by the nearest neighbor lookup in the k-d tree. By doing so, the time complexity of matching drops to $O(N \log(N))$. Subsequently, this set of putative matches is used to estimate the transformation. However, the matching process is not well-conditioned in the sense that there are inevitably some keypoints that cannot be matched, and for some others multiple matches with the same similarity are found. This yields many spurious matches in the initial set (it is not uncommon that fewer than 25% are correct). Common estimation techniques — least squares, least median, or M-estimators — perform badly with this large fraction of outliers [122]. Instead, RANSAC (RANdom SAmple Consensus) [20] or RANSAC-derived methods [138, 147] are often used to eliminate mismatches and to robustly estimate

the transformation.

It is worth noting that the optimization is carried out by minimizing the geometric distance between matched points, as shown in Equation 2.1. Therefore, the methods based on keypoint extraction and matching are still considered as feature-based methods. One may argue that the matching between keypoints is conducted using the invariant descriptor, which is a similarity measure based on local intensities. However, because the objective function does not depend on intensity differences, these methods do not belong to the intensity-based category.

Methods based on keypoint extraction followed by similarity matching in the feature space have been applied to 2d image registration and mosaic construction [23], fundamental matrix estimation [134], object recognition [122], location recognition [60], 3d registration [183] and simultaneous localization and mapping [183].

2.1.2 Intensity-based methods

The history of intensity-based registration methods dates back to 1970s [9]. In 1981, Lucas and Kanade proposed an optical flow registration algorithm [124], which has become one of the most commonly cited computer vision papers. 20 years later, Baker presented an overview of the Lucas-Kanade algorithm, its extensions, and experimental comparisons [8].

These methods aim at finding a transformation function such that after the application of the transformation function, the photometric differences between corresponding pixels (or voxels) are minimized. These methods often work in an iterative manner: warp one image on to the coordinate system of the other using the current transformation estimate, compute an update of the transformation from the image gradients or optical flow vectors, and repeat these steps until the transformation estimate converges.

A simple and canonical measure used in intensity-based methods is the Sum of Squared Difference (SSD) between the warped image and the fixed image, presented by Barnea and Silverman in 1972 [9] and later on extended to achieve subpixel

accuracy [215]. The objective function using SSD measure is defined as

$$\sum_{\mathbf{x}} \left[I_q(\mathbf{T}(\mathbf{x};\boldsymbol{\theta})) - I_p(\mathbf{x}) \right]^2 , \tag{2.2}$$

where $\mathbf{T}(\mathbf{x};\boldsymbol{\theta})$ is the transformation function that maps coordinates of I_p onto I_q as mentioned above.

This objective function is non-linear because it involves the image functions I_p and I_q that map coordinates to intensities. To solve this non-linear problem, the minimization is often carried out by gradient descent methods in the following way. Assuming the current estimate is $\hat{\boldsymbol{\theta}}$, Equation (2.2) can be linearized by performing a first order Taylor expansion on $I_q(\mathbf{T}(\mathbf{x};\boldsymbol{\theta}))$ around $\hat{\boldsymbol{\theta}}$ and applying chain rules on derivatives:

$$\sum_{\mathbf{x}} \left[I_q(\mathbf{T}(\mathbf{x};\hat{\boldsymbol{\theta}})) + \nabla I_q \frac{\partial \mathbf{T}}{\partial \boldsymbol{\theta}} \Delta\boldsymbol{\theta} - I_p(\mathbf{x}) \right]^2 , \tag{2.3}$$

Note that in this Taylor expansion, only the update $\Delta\boldsymbol{\theta}$ is unknown and everything else is known. Thus, it becomes a normal linear least-squares problem and the update solved accordingly [8, 62].

These methods require good initialization since the capture range is very small — typically a few pixels. It has been shown that with multi-resolution settings (usually from coarse to fine) [14] these methods are more effective and have a broader capture range. In addition to multi-resolution, a initialization method using a coarse search on translations has been shown adequate in some cases [179]. However, no known intensity-based initialization methods starts with affine or more complex transformation models due to the high-dimensional search space.

SSD is sensitive to intensity variations, such as shutter speed change or illumination changes. As can be seen from the objective function defined in Equation (2.2), photometric changes in one image would alter the shape of the objective function and affect the minimization accordingly. Several modifications have been proposed to cope with these situations. For example, Black and Anandan proposed a robust M-estimator instead of the usual square function [18]. Also, a normalized SSD measure, proposed in the early 80's, uses pixel intensities in a small neighborhood to normalize the intensity magnitudes such that the measure is invariant to

scaling of intensities [79]. Denoting $\mathcal{R}$ as a local region around a given pixel $\mathbf{x}$ and $\mathbf{y}$ as the displacement vector inside region $\mathcal{R}$, the objective function using normalized SSD measure is given by

$$\sum_{\mathbf{x}} \frac{\sum_{\mathbf{y}\in\mathcal{R}} \left[I_q(\mathbf{y}+\mathbf{T}(\mathbf{x};\boldsymbol{\theta})) - I_p(\mathbf{y}+\mathbf{x})\right]^2}{\sum_{\mathbf{y}\in\mathcal{R}} \sqrt{I_q(\mathbf{y}+\mathbf{T}(\mathbf{x};\boldsymbol{\theta}))^2 I_p(\mathbf{y}+\mathbf{x})^2}} \tag{2.4}$$

Similarly, a measure of normalized cross-correlation was developed to achieve invariance to shift and scaling of intensities [68].

Intensity-based methods suffer from several drawbacks: they are susceptible to local minima and outliers and are also unnecessarily inefficient [191]. A simple example for the inefficiency can be drawn from pixels in a large homogeneous region (i.e. a region with one uniform color): image gradients in such a region, small in magnitude and severely affected by noise, have little (perhaps even adverse) use in the transformation estimation. A modified approach is to use local patches of interest in the estimation [14, 191]. The choices for the interest can be based on intensity variation, such as corners or region with rich texture [216], or positive definite Hessian matrix [93], but are not limited to these.

In particular, Irani proposed to compute and normalize derivatives in four directions as the "intensity value" at each pixel and then the registration was carried out by maximizing normalized cross-correlation on these intensities (derivatives) [93]. This method has been successfully applied to register two multimodal image pairs.

2.1.3 Mutual information

Mutual information registration was developed independently by Wells and Voila [233] and by Maes and Collignon [127]. Since then mutual-information-based image registration has been an active field of research — more than 400 published papers so far — and has become common practice in many clinical applications. Readers are referred to a survey paper [165] for more details.

First we introduce the notion of entropy, originated from communication theory. Entropy of a random variable can be viewed as a measure of *uncertainty* —

the higher entropy, the more uncertain the random variable behaves. Denoting a random variable as P and probability of event e_i as p_i, the commonly used Shannon entropy is defined as

$$\mathrm{H}(P) = -\sum_i p_i \log p_i.$$

When two images, I_A and I_B, taken from different modalities are well aligned, intensities of corresponding points exhibit some unknown functional relationship, which is reflected by a joint probability distribution of the two intensity values. It is suggested to use entropy as a measure of alignment, because "entropy measures the dispersion of a probability distribution. It is low when a distribution has a few sharply-defined dominant peaks and it is maximal when all outcomes have an equal chance of occurring" [165]. A joint histogram is used to estimate this joint probability distribution, computed from intensity values on I_A and I_B. Dividing the entries of the histogram by the total number of entries yields a density distribution, denoted as $p(a, b)$. The Shannon entropy for this probability joint distribution is defined as

$$\mathrm{H}(A, B) = -\sum_{a \in A,\ b \in B} p(a, b) \log p(a, b), \tag{2.5}$$

where A and B denote the set of possible intensity values on image I_A and I_B, respectively. Note that when p_A and p_B are independent, $\mathrm{H}(A, B) = \mathrm{H}(A) + \mathrm{H}(B)$, whereas when p_A and p_B are perfectly correlated $\mathrm{H}(A, B) = \mathrm{H}(A) = \mathrm{H}(B)$.

The mutual information is the relative entropy between the joint distribution and the product distribution. Given two images I_A and I_B, the mutual information can be defined as

$$\begin{aligned} &= \mathrm{H}(A) + \mathrm{H}(B) - \mathrm{H}(A, B) \\ &= \mathrm{H}(B) - \mathrm{H}(B|A) \\ &= \mathrm{H}(A) - \mathrm{H}(A|B). \end{aligned} \tag{2.6}$$

After the application of a transformation estimate, the mutual information is computed only in the overlap area. The goal of image registration is to find a transformation estimate such that it maximizes the mutual information.

Compared to the joint entropy, mutual information has the advantage that it includes the entropies of separate images. Because the mutual information and joint entropy are computed only in the overlap area, they are sensitive to the contents in the overlap. When using joint entropy alone, a problem may occur that small joint entropy value is found for complete mis-registration. For instance, an image registered to a homogeneous region results in low joint entropy. In contrast, because the mutual information includes the marginal entropies, it is more likely to avoid this problem [165].

Since the publication of the two original mutual information papers [127, 233], there has been a considerable amount of development in this area, some of which are worth mentioning.

- Several normalized mutual information measures [202] have been proposed to eliminate bias toward low overlap. Studholme *et al.* proposed $\mathrm{NMI}(A, B) = (\mathrm{H}(A) + \mathrm{H}(B))/\,\mathrm{H}(A, B)$ [202], whereas Collignon and Maes suggested the use of Entropy Correlation Coefficient (ECC) [127]. The two are equivalent in the following manner: $\mathrm{ECC} = 2 - 2/\,\mathrm{NMI}$.

- A drawback of mutual information based on Shannon entropy is that the dependency of neighboring pixel intensities is ignored. Kybic proposed to use KL entropy estimator to incorporate neighboring pixel intensities as well as multiple channels per pixel [110].

- Pluim *et al.* proposed to incorporate gradient information with mutual information [164], to achieve a smoother objective function and hence broaden the domain of convergence. Gradient information can be applied to register multimodal images because "the images fundamentally depict the same anatomical structures, gradients in two multimodal images — at least in principle — will have the same orientation and either identical or opposing directions." [164] As we will see in Chapter 5, this is similar to how GDB-ICP achieves multimodal image registration.

Optimization is usually carried out by two categories of approaches: non-derivative-based and derivative-based minimization methods. For the first category,

one popular choice is Powell's method, which optimizes each transformation parameter in turn [115, 127, 96]. Another popular choice is the simplex method, which considers all parameters simultaneously [128, 188]. For the derivative-based approaches, the exact expression for the gradient calculation on mutual information was proposed in [128], with details on gradient ascent, quasi-Newton, and Levenberg-Marquardt methods.

The strength of mutual information lies in that fact that it does not presume any functional relationship between intensities on two images. The intensity relationship is not known until it is estimated during the image registration process. Thus, mutual information has a broad range of applications and can handle a wide variety of imaging modalities. However, for the very same reason, the optimization is sensitive to the overlap area and is subject to multiple local minima. Therefore, the initialization must be close for it to converge to the correct transformation. Furthermore, it turns out that mutual information is not well applicable to images with long and thin structures such as retinal images [96], or to the combination of CT and ultra-sound images [174].

2.2 Transformation models

The appropriate use of transformation models depends upon the physical imaging process. For instance, when aligning two 3d LADAR scan datasets, a 3d rigid model — rotation and translation — is enough [15]. When aligning two 3d CT lung volumes, not only a 3d rigid model, but also a deformable model is needed to accommodate the respiration process [44]. In general, transformation models can be categorized into parametrized and non-parametrized models. Within parametrized models, there are global models (parameters do not depend on location) and local or deformable models. For more details, readers are refer to image registration survey papers [22, 244] and a transformation model survey paper [67].

2.2.1 Global models

Global models are often referred to as those that have only a small number of parameters that describes the mapping of the entire image.

2.2.1.1 2d-to-2d models

For aligning 2d images that are projections of a 3d scene, a variety of models are available from lower order to higher order: translation, rigid, similarity, affine, and homography [180]. The translation model is given by

$$\mathbf{q} = \mathbf{p} + \mathbf{t}.$$

The rigid model is

$$\mathbf{q} = \mathbf{R}\mathbf{p} + \mathbf{t},$$

where rotation matrix, $\mathbf{R}$, is an orthonormal matrix. In 2d, $\mathbf{R}$ is defined with one rotation angle θ:

$$\mathbf{R} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}.$$

Allowing scaling, the similarity model is

$$\mathbf{q} = s\mathbf{R}\mathbf{p} + \mathbf{t},$$

where s is the scaling factor. The affine model is given by

$$\mathbf{q} = \mathbf{A}\mathbf{p} + \mathbf{t}.$$

The affine model in general does not preserve orthogonality, i.e., a right angle is no longer a right angle after application of the affine transformation.

Let the homogeneous coordinates of $\mathbf{p}$ be $\tilde{\mathbf{p}}$ and the homogeneous coordinates of $\mathbf{q}$ be $\tilde{\mathbf{q}}$. The homography model can be written as

$$\tilde{\mathbf{q}} \tilde{=} \mathbf{H}\tilde{\mathbf{p}},$$

where $\tilde{=}$ means equivalence up to a non-zero scale. After application of the homography transformation, parallel lines may intersect at a finite point — the homography transformation does not preserve parallelism. Though the homography model is

linear in homogeneous coordinates, the model is nonlinear in Euclidean space:

$$\begin{pmatrix} q_x \\ q_y \end{pmatrix} = \frac{1}{h_{31}p_x + h_{32}p_y + h_{33}} \begin{pmatrix} h_{11}p_x + h_{12}p_y + h_{13} \\ h_{21}p_x + h_{22}p_y + h_{23} \end{pmatrix} \tag{2.7}$$

Note that the above models are nested in nature: the affine model is a subset of the homography model, the similarity model is a subset of the affine model, and so on. For a more detailed and complete description about the 2d-to-2d transformation models, readers are referred to [82, Ch. 1] and [22]. We do not consider the fundamental matrix as a transformation model because for any given point, the corresponding point can be anywhere along the epipolar line.

The above 2d models have assumed the camera is a perfect pinhole camera. However, this assumption will not hold in practice. Geometric and photometric distortions may appear in images acquired by cameras, especially off-the-shelf ones. Specifically, the geometric distortion, often called lens distortion, has two major types: radial and tangential [82, Ch. 6]. In practice, radial lens distortion is often dominant and generally increases as the focal length decreases.

Denote the undistorted image point as $\mathbf{p}_u$ and the distorted image point as $\mathbf{p}_d$. Let $\mathbf{p}_0$ be the center of the distortion and $\mathrm{r}^2(\mathbf{p}) = \|\mathbf{p} - \mathbf{p}_0\|^2$ be the squared radial distance function. the undistorted-to-distorted 4th-order formulation is

$$\mathbf{p}_d = (\mathbf{p}_u - \mathbf{p}_0)\left(1 + \kappa_1\,\mathrm{r}^2(\mathbf{p}_u) + \kappa_2\,\mathrm{r}^4(\mathbf{p}_u)\right) + \mathbf{p}_0.$$

And the distorted-to-undistorted 4th-order formulation is

$$\mathbf{p}_u = (\mathbf{p}_d - \mathbf{p}_0)\left(1 + \kappa_1\,\mathrm{r}^2(\mathbf{p}_d) + \kappa_2\,\mathrm{r}^4(\mathbf{p}_d)\right) + \mathbf{p}_0.$$

κ_1 and κ_2 are the 2nd-order and 4th-order radial distortion term coefficients, respectively. It has shown by Tamaki that both formulations are equally-effective approximations to the real distortion [210]. Sawhney [180] and Meng [139] uses only the 2nd-order radial distortion term, whereas Stein uses also the 4th-order term [198]. Photometric distortion is mostly caused by vignetting and compensations have to be made for some sensitive applications [6].

There are other transformation models besides these commonly-used ones. In particular, Can *et al.* derived a 12-degree-of-freedom quadratic model from a combination of weak perspective camera model and the quadratic retina surface:

$$\begin{pmatrix} q_x \\ q_y \end{pmatrix} = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \theta_{14} & \theta_{15} & \theta_{16} \\ \theta_{21} & \theta_{22} & \theta_{23} & \theta_{24} & \theta_{25} & \theta_{26} \end{pmatrix} \begin{pmatrix} p_x^2 \\ p_x p_y \\ p_y^2 \\ p_x \\ p_y \\ 1 \end{pmatrix}$$

This model achieves sub-pixel accuracy in the registration of retinal images [32]. The same quadratic model is also used in the mosaic formation of 2d images [108] and in stereo problems [126, 238].

2.2.1.2 3d-to-3d models

For 3d images, the commonly-used models are rigid, similarity, and affine [15, 103]. Their formulations are the same as the corresponding 2d models except the dimension difference.

2.2.1.3 3d-to-2d models

Let the 3d scene point be $\mathbf{P}$ and the 2d image point be $\mathbf{q}$. Let their homogeneous coordinates be $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{q}}$. Assuming the camera is a pin-hole camera, The perspective camera projection in homogeneous coordinates is given by

$$\tilde{\mathbf{q}} \tilde{=} \mathbf{C}\tilde{\mathbf{P}},$$

where the 3×4 matrix $\mathbf{C}$ is the projection matrix. It is known the projection matrix $\mathbf{C}$ can be decomposed into intrinsic and extrinsic parameters:

$$\mathbf{C} = \mathbf{KR}\left[\mathbf{I}_{3\times 3} | - \mathbf{P}_c\right], \tag{2.8}$$

where $\mathbf{P}_c$ is the location of the camera, $\mathbf{I}_{3\times3}$ is the 3x3 identity matrix, $\mathbf{R}$ is the rotation matrix, and $\mathbf{K}$ is the upper-triangular calibration matrix. The calibration matrix $\mathbf{K}$ encapsulates all the intrinsic parameters:

$$\mathbf{K} = \begin{pmatrix} \alpha_x & k & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{pmatrix},$$

where α_x is the scaling factor along the x axis, α_y is the scaling factor along the y axis, k is the skew, and $[u_0, v_0]^\top$ is the principal point.

For a perfect pin-hole camera, the skew is zero, α_x is equal to α_y (square pixel), and the principal point is at the center of the image. However, due to modeling errors[4] and manufacture imperfections, the above statements only approximately sustain. For instance, the real skew is a non-zero value, but rather small when compared to the α_x and α_y.

In addition, lens distortion must be accounted for when seeking an accurate projection. The lens distortions are discussed in the context of 2d-to-2d models in Section 2.2.1.1.

For a more detailed description and a complete list of camera models, readers referred to [82, Ch. 5].

2.2.2 Local/Deformable models

Global transformations cannot properly handle images that have local deformations. This happens often in medical imaging where the subject is continuously undergoing some deformations, such as respiration cycles. Deformable models have two categories: parametrized and non-parametrized. Readers are referred to surveys in [135] and [176, Ch. 13] about the use of deformable models in medical imaging.

Parametrized deformable models are generally defined in terms of basis functions [213, 212], with coefficients forming the parameters to be estimated. Compared to the small number of parameters in global models, the number of parameters is large, often proportional to image resolution and dimension. Basis functions can

[4] Cameras are not really pin-hole. They are approximately pin-hole when the aperture is small.

have either infinite support or finite support. One example of infinite support is Fourier basis functions [4]. The disadvantage of infinite support is that a change on a single parameter affects the whole transformation. It is often undesirable because this global influence makes it difficult to model and to adjust deformations locally in isolation. Hence, it is not a surprise that local-support basis functions are more popular in deformable image registration.

Local-support basis functions have non-zero values only in a small region and are zero everywhere else. Various basis function have been proposed for the use in the image registration, including piecewise linear models [70], piecewise cubic models [71], cubic B-splines [175, 177, 214], and truncated Gaussian kernels [189] (more details in [244]).

There are also other parametrized deformation functions not using basis functions. In particular, deformations are defined local/piecewise affine transformations [57, 97, 161, 163, 239]. Each of the affine transformations is estimated and only applicable to a small region. The deformations between regions are computed by (linearly) interpolating the affine transformations. This approach has the advantage that the overall transformation is linear and easy to estimate.

Lastly, local deformations can also be modeled using implicit functions, such as displacement fields. The registration is then formulated with partial differential equations and can be solved by finite element methods [87, 86].

2.3 3d scene to 2d image registration

According to the scheme shown in Figure 2.1, image registration methods can also be classified according to the spatial dimension of image and scene. One category is 3d-to-2d (scene-to-image) registration.

3d-to-2d registration has a wide variety of application, spreading across several areas — camera calibration [5, 234, 204, 224, 241], location recognition and camera pose estimation [95, 111, 123, 173], and simultaneous localization and mapping [60, 66, 77, 158, 183].

As shown in Section 2.2.1.3, the perspective camera matrix can be decomposed

into intrinsic and extrinsic parameters[5] (Equation 2.8). Camera calibration methods generally focus on the estimation of the intrinsic parameters, often including lens distortion terms. By contrast, camera pose and location recognition techniques often presume the camera is pre-calibrated or the camera intrinsic parameters are known a priori and estimate only the extrinsic parameters [119, 123, 158, 173]. There are also methods that emphasize the estimation of both [235, 237].

2.3.1 Structural Features

One type of 3d-to-2d registration method is based on extraction and matching of structural features, such as line segments, between the model and the image. This has been used in urban environments for refining an initial camera position estimate relative to a 3d model [120, 194]. Vanishing points detected from parallel lines are used in [173] to estimate a hand-held camera location and orientation with respect to a database of views of building facades. Similarly, Liu and Stamos [119] extract line segments from the model and the image, compute three sets of major vanishing points to determine the intrinsic parameters and the orientation, and cluster line segments to form a parallelepiped to determine the translation. However, since these methods require detection of at least two sets of orthogonal parallel lines and calculation of the intersection points, due to noise contamination and pixel discretization, the estimation of intrinsic parameters can be unstable and error-prone, resulting in inaccurate results.

In addition to geometric features, Troccoli and Allen compute shadows from the range date using the prior knowledge about the position of the sun in the sky. They estimate the camera pose by matching the shapes of shadows detected in the image and the shapes of computed shadows projected onto the image. The optimization is carried out by simulated annealing.

The above methods extract features, e.g. line segments, from the range data to drive the registration. Besides range data, Umeda, Godin, and Rioux also employ laser reflectivity data and estimate the pose by maximizing the similarity between the image and the projected laser reflectance image [227].

[5]The camera extrinsic parameters are often referred to as “camera pose”.

Due to the high dimension of the search space and the difficulty of the optimization, this type of methods often presumes a simplified camera model and no lens distortion. As the result, these methods are most often seen applied to camera pose estimation or location recognition problems.

2.3.2 SLAM

Simultaneous Localization And Mapping (SLAM) is a technique used in robotics to construct a map of an unknown environment while at the same time keeping track of robot location. Robot positioning is difficult because of the inherent uncertainties in determining the robot's relative movements from its sensors [151]. In our discussion, we focus on robots equipped with cameras and using images to find its location.

Se, Lowe and Little [183] extract SIFT keypoints [122], back-project them into 3d using a trinocular stereo vision system, and match them against keypoints stored in the 3d scene model. Matches from different keypoints are combined using RANSAC and Hough methods to estimate the location of the robot. Newman, Cole and Ho [158] match subsequences of images taken by a mobile robot equipped with both intensity cameras and range scanners in order to solve the "loop closure" problem. Keypoints are extracted and clustered as the basis for matching image subsequences. Techniques are introduced for handling repetitive appearance, and the relative pose is solved using image-based techniques for initialization and the associated range data for refinement. Gutmann and Konolige [77] exploit odometry to initialize 3d-to-3d registration in solving the loop-closing problem. Several systems have been developed to exploit the abundance of predominately linear features in a urban environment to supplement GPS information [66] or to solve the "kidnapped robot" problem — identifying the location of the robot without prior knowledge of its position [173]. Recently, Fraundorfer and Bischof [60] addressed the "kidnapped robot" problem by matching an image against a piecewise planar (indoor) scene model, initializing based on a single keypoint match, confirming through correlation, and then estimating pose.

2.3.3 Camera Calibration

Camera calibration is the process of determining the intrinsic and extrinsic camera parameters. It is closely related to the estimation step in image registration techniques — the parameter estimation uses feature correspondences formed between the 3d model and the 2d image. Unlike image registration, the matching process can be either automatic or manual. Another difference is that the calibration images are often taken with calibration targets placed in the scene. Other techniques — "self calibration" methods — calibrate a camera without the presence of a calibration target, but require multiple views [56, 130, 167]. Although self-calibration is outside the scope of this thesis, a survey of self calibration can be found in [85].

Planar calibration targets are widely used in camera calibration [5, 204, 224, 241]. Comparative reviews with accuracy evaluation can be found in [5, 38]. Zhang [241] and Sturm [204] separately-but-simultaneously suggested estimating a simplified camera from plane homographies using orthogonality constraints. Since each homography has eight degrees of freedom and the camera extrinsic parameters have six, each homography provides two constraints on the intrinsic parameters. As a result, one homography can be used to compute the scaling factors of x and y axes while holding other intrinsic parameters as fixed. Sturm and Maybank carefully studied the singular configurations of this method where multiple solutions exist for one configuration [204]. Tsai proposed a two-step method which first computes the 3d rotation and translation on the x and y axes using displacement vectors originated from the center of the image and they during the second step computes the focal length, distortion coefficients, and translation on the z axis [224]. During the first step, this method assumes the principal point is at the center of the image and there is no skew. All parameters are iteratively optimized after the two steps. Weng also proposed a two-step method alternating between computation of the intrinsic and extrinsic parameters and computation of the radial, tangential, and thin-prism distortion coefficients [234].

Several camera calibration papers estimate the camera instrinsic parameters radial lens distortion coefficients by straightening lines in the image [2, 50]. Similarly,

Caprile and Torre estimate the camera intrinsic parameters using the vanishing points of parallel lines in the image to determine the image of the absolute conic [35]. However, since they do not assemble image registration methods, we will not focus on these methods.

2.3.4 Medical Domain

There is a large volume of papers published in medical imaging on 3d-to-2d registration. For instance, Cyr *et al.* presented a method to register a 3d CT volume with a target 2d X-ray view by minimizing the difference between the target X-ray view and a constructed view of the 3d CT volume data [46]. In a similar 3d-to-2d vascular image registration work, Lau and Chung matched the digital reconstructed radiograph of the 3d model with the target 2d image [111]. To avoid local minima, they systematically generated sample poses and applied Powell's optimization method at different resolution levels. In order to automate the navigation of a surgical robot, Iwashita *et al.* applied steepest descent on a distance map to compute the object pose that best aligns the 3d model and the target 2d image without external markers or special measurement devices [95]. Finally, McLaughlin *et al.* conducted a comparison study between a intensity-based method and a feature-based method for 3d-to-2d registration in the context of neurointerventions concluding that the intensity-based method is more accurate while the feature-based method has a wider capture range and is less expensive computationally [136].

2.4 Multi-image registration

The above discussion has focused on the registration of two images. In many cases, the registration of a set of images (larger than two) is desirable for applications ranging from panorama formation [23, 180, 191], super resolution [34, 187], to (temporal) change detection [172].

2.4.1 Bundle Adjustment

The goal of multi-image registration is to compute a set of globally-consistent transformations between all pairs of images [191, 23]. Simple approaches include

registering images to the panorama one at a time [209], or, when dealing with a video sequence, aligning each frame to the previous one [94]. However, error accumulates along the process and misalignments appear between images. For instance, when closing a loop using a cylindrical model, the last image does not overlay with the first one consistently. It is because the transformations obtained locally between images are not accurate enough to form a globally-well-aligned panorama.

The solution is to distribute errors evenly across all overlapping images, using a technique called *bundle adjustment*, which first appeared in photogrammetry. This technique simultaneously adjusts the transformation parameters for all overlapping images [220]. It was first applied to general structure-from-motion problems in computer vision [208], then later to multi-image registration [180, 191]. Bundle adjustment is formulated using correspondences from overlapping image pairs — a feature-based approach. Though it is possible to simultaneously minimize photometric differences between all overlapping image pairs [180], it is computationally too expensive as it involves all overlapping pixels and re-warping these pixels at every iteration. Using sparse feature correspondences is much more efficient and is the path often taken [23, 34, 192, 196, 207].

2.4.2 Panorama Construction

The construction of a panorama is worth mentioning as it is one of the most popular applications that require image registration of multiple images. The major steps of panorama construction process include registering image pairs, extracting correspondences from the registered pairs, jointly estimating the transformation aligning all images (bundle adjustment), and constructing the actual panorama by transforming images and blending pixel intensities. The final step can be as easy as computing an intensity average at each pixel. However, this usually does not work very well, due to vignetting, exposure differences, mis-registration, scene movement, or other artifacts that would corrupt in the final mosaic. There are two important aspects to be considered: seam selection and the blending technique [207]. To select an optimum seam between images, dynamic programming [47], graph-cut [1], and vertex cover [228] methods have been applied to calculate a seam that has minimum

differences between two images. The actual blending can be done using a weighted average based on a distance map (feathering) [228], a band-pass Laplacian pyramid [23, 28], wavelets [205], or gradient domain blending [1].

2.5 Summary

This chapter has summarized image registration methods with an emphasis on the to-be-optimized metric: 1) feature-based methods, 2) intensity-based methods, 3) mutual-information-based methods. These methods were discussed in the context of 2d-to-2d pairwise image registration. In addition, we discussed 3d-to-2d image registration methods and multi-image registration methods aiming at aligning multiple images.

CHAPTER 3
Automated Masking

3.1 Overview

Automatic masking is an important process for retinal image analysis, especially for retrospective studies in which images are digitized from slides.

Our approach exploits a variety of techniques in the literature, including the active contours, dynamic programming, K-means clustering, and edge detection. Active contours (snakes) were first introduced by Kass in 1988 [100]. In 1990, Amin [3] proposed exploiting dynamic programming for the optimization of active contours and claimed that this approach resulted in the global optimum within the discretized search window. Among the many techniques that have been proposed for edge detection [21, 83, 105], Canny's method [33] is among the most commonly used. K-means classification algorithm, presented in detail by Hartigan [81], has been widely used in many applications.

There are two major steps in our method: detection and refinement. Detection is carried out on a coarse scale of the image by generating a set of mask boundary candidates and locating the one that best separates the background from the retina surface. Following detection, refinement is applied in a coarse-to-fine manner, using an active contour technique. The final boundary is smoothed and a mask image is produced. The method is outlined in Figure 3.1 and is demonstrated step by step in Figure 3.2.

In the experiments, we use a development set of 29 retinal images and then evaluate the algorithm on a set set over 2,000 images. We also show quantitatively the accuracy of automatically generated mask images when compared to the known ground truth mask.

3.2 Detection

Most of the difficulty in automatic masking lies in detecting the (approximate) location of the mask boundary. As we have observed in Figure 1.6, the masks do

1. **Pre-processing**
   (a) Build a Gaussian image pyramid
   (b) Calculate intensity gradients at each level of the pyramid
2. **Detection**
   (a) Apply the modified K-means
   (b) Apply Canny edge detection
   (c) Combine the K-means result, the edges, and the gradient image to produce *verified points*.
   (d) Form the 1st mask boundary candidate based on the *verified points*.
   (e) Link the Canny edgels chains to form other mask candidates.
   (f) Compute the objective function value (Equation 3.4) for each candidate.
   (g) Choose the one candidate that has the minimum value of the objective function.
3. **Refinement**
   - Repeat
     (a) Refine the current boundary by applying an active contour technique.
     (b) Propagate the boundary to a finer resolution
   - Until it reaches the original image resolution.
4. **Contour smoothing and mask image generation.**

Figure 3.1: Outline of the automatic masking

not have a fixed shape. Also, the outmost boundary may be incorrect due to the scanning process. The mask area is not entirely dark and homogeneous. We make an assumption that the mask is a convex shape (excluding the upper-right tag). Restricting the mask to a convex shape is useful when dealing with images where the intensities of retina pixels are close to those of the mask region, such as the example shown in Figure 1.6a and Figure 1.7a.

The detection of mask boundary contour is based on a combination of a mod-

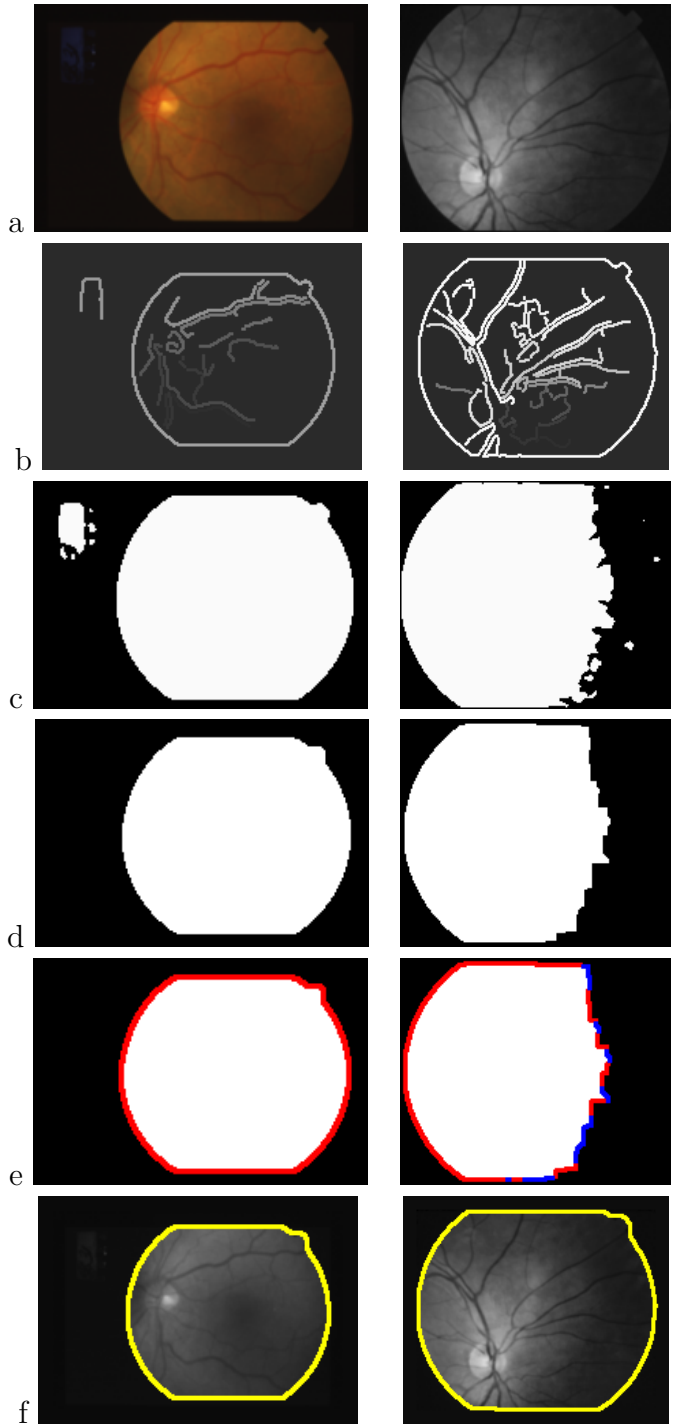


Figure 3.2: Demonstration of the automatic masking method: a) input images, b) the result of Canny edge detection, c) the segmentation result, d) after applying the connected components on the segmentation labels, e) the verified boundary points (red are verified and blue are failed), f) the boundary points of the best candidate

ified intensity-based K-means segmentation method and the Canny edge detection method, applied at the coarsest resolution of a Gaussian pyramid. Intensity and edge information is used in a combined way because neither one is sufficient: without edge information, dark retina area may be indistinguishable from the mask (see the example shown in Figure 1.6a and Figure 1.7a); whereas without intensity information, the mask boundary may be broken into edge segments and indistinguishable from other edges (Figure 3.3). Combining both information gives a higher detection rate (details are discussed in Section 3.4).

3.2.1 Modified K-means

First, a modified version of K-means is applied to divide the image into two clusters: the foreground (the retina) and the background (the mask). It takes the intensity of each pixel and alternates between 1) computing the statistics within each cluster (estimation step) and 2) re-clustering the pixels according to the distance calculated using the statistics (labeling step). The use of K-means comes from the observation that the foreground is a single region, generally bright, high in contrast, and located around the center of the image, whereas the background, when compared to the foreground, is generally darker and homogeneous and located on the periphery of the image.

We compute the robust mean and variance for each cluster and use Mahalanobis distance for the labeling. Robustness is needed because there are artifacts in the image and because the intermediate steps may contain incorrect labels. For color images, we assume the covariance matrix is a diagonal matrix, i.e., RGB channels are independent. Thus, the mean and variance are estimated separately for each channel. The overall distance is a sum of the Mahalanobis distances computed from each of the three channels.

During the estimation step, an M-estimator is used to compute the mean and variance. As there is no closed-form solution, we apply an Iterative Re-weighted Least Squares (IRLS) method [74, 137, 200]. Denote the image as a function $\mathcal{I}(x, y)$. At a floating point location the intensity is linearly interpolated. Define the label $l \in \{b, f\}$ as background, b, or foreground, f. Then the set of pixels in the foreground

is $\mathcal{C}_f$ and the set of pixels in the foreground is $\mathcal{C}_b$, , the estimation of the mean and the variance for a cluster l (either $l = b$ or $l = f$) and a channel c is carried out by:

$$\begin{aligned} \mu_{l,c} &= \frac{\sum_{j\in\mathcal{C}_l} w^r_{l,c}(x_j, y_j;)w^s_l(x_j, y_j)I_c(x_j, y_j)}{\sum_{j\in\mathcal{C}_l} w^r_{l,c}(x_j, y_j)w^s_l(x_j, y_j)} \\ \sigma^2_{l,c} &= \frac{\sum_{j\in\mathcal{C}_l} w^r_{l,c}(x_j, y_j)w^s_l(x_j, y_j)(I_c(x_j, y_j) - \mu_{l,c})^2}{\sum_{j\in\mathcal{C}_l} w^r_{l,c}(x_j, y_j)w^s_l(x_j, y_j)}, \end{aligned} \tag{3.1}$$

where $w^r_{l,c}(x_j, y_j)$ is the standard M-Estimator weighting using the estimates from the previous IRLS iteration:

$$w^r_{l,c}(x, y) = \frac{\rho'((I_c(x, y) - \mu_{l,c})/\sigma_{l,c})}{(I_c(x, y) - \mu_{l,c})/\sigma_{l,c}}. \tag{3.2}$$

$\rho(u)$ is the Beaton-Tukey robust loss function [10]

$$\rho(u) = \begin{cases} \frac{a^2}{6}[1 - (1 - (\frac{u}{a})^2)^3], & |u| \le a \\ \frac{a^2}{6}, & |u| > a \end{cases} \tag{3.3}$$

$w^s_f(x_j, y_j)$ and $w^s_b(x_j, y_j)$ are radial weighting functions computed from the radial distance from the image center, (x_c, y_c):

$$w^s_f(x, y) = 1 - \frac{r(x, y)}{r_{max}} \qquad w^s_b(x, y) = \frac{r(x, y)}{r_{max}}$$

where $r(x, y) = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ and $r_{max} = \max_{(x,y)\in\mathcal{I}} r(x, y)$.

IRLS alternates between the robust weight $w^r_{l,c}(x, y)$ computation and the estimation of the mean $\mu_{l,c}$ and the variance $\sigma^2_{l,c}$ until convergence. The computation is repeated for each channel and for each cluster.

The next step is to "label" pixels using Mahalanobis distance, which is summed across all channels: $d^2_f(x, y) = \sum_c \frac{(I_c(x,y)-\mu_{f,c})^2}{\sigma^2_{f,c}}$ and $d^2_b(x, y) = \sum_c \frac{(I_c(x,y)-\mu_{b,c})^2}{\sigma^2_{b,c}}$. In K-means, a pixel is labeled to the cluster by which its distance is minimized. For instance, if $d^2_f(x, y) > d^2_b(x, y)$, the pixel is classified as foreground, while if $d^2_f(x, y) \le d^2_b(x, y)$, the pixel is background.

To get the modified K-means started, initial statistics are computed from pre-

defined regions: pixels along the perimeter of the image are background, whereas pixels inside a small 20x20 box placed at the image center are foreground. Then the modified K-means method iterates until the labels no longer changes. The results are shown in Figure 3.2c.

After convergence, the result needs to be cleaned in order to joint the dark vessels to the foreground and remove artifacts in the background. A morphological closing operator with a 3×3 square kernel is applied to the label image followed by connected-components on the labels. The resulting label image often resembles the mask image in the coarse scale (the left column of Figure 3.2c and also Figure 3.3d). But in some cases, this label image is incorrect (the right column of Figure 3.2c). The latter indicates the need to exploit other information, such as edges.

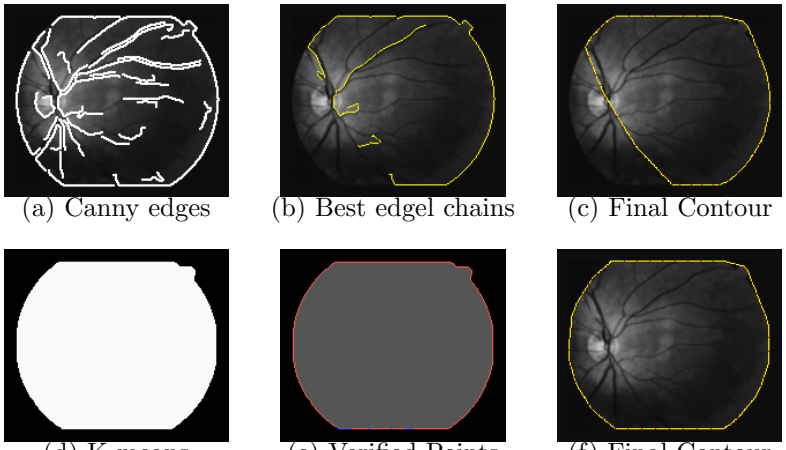
3.2.2 Edge Detection

Edge detection is carried out by the commonly-used Canny edge detector [33], which finds points with large gradient magnitudes, applies non-maxima suppression, groups the points into *edgel chains* using hysterisis, and prunes short edgel chains. The results of Canny edge detection are shown in Figure 3.2b and in Figure 3.3a. The readers should pay attention to Figure 3.3, where the mask boundary is broken into several edgel chains and linked with vessel boundaries. This indicates that we cannot solely rely on edge detection to get the mask.

3.2.3 Verified Points

The modified K-means does not always produce satisfactory labeling. However, even when it fails, many labels are still correct. Defining the "boundary points" as the foreground pixel whose at least one of the 4-connected neighbors labeled as background, we hence propose to *verify* these boundary points.

For each of the boundary points, the verification process examines pixels in a 3x3 window with two criteria: 1) the pixel is an edge point and 2) the directional derivative projected onto the normal direction — normal to the boundary contour — is larger than a threshold (empirically defined as 2) and is no less than 1/2 (also empirically defined) of the gradient magnitude. A boundary point is a *verified* point if both criteria are met for at least one point within the 3x3 window. Intuitively, the



(a) Canny edges (b) Best edgel chains (c) Final Contour

(d) K-means (e) Verified Points (f) Final Contour

Figure 3.3: An example where the modified K-means and verified points are better than grouping of edgel chains. (a) shows the Canny edgel chains. (b) shows in yellow the best edgel chain groups that achieves the minimum objective function value. (c) shows in yellow the resulting piecewise linear contour from the best edgel chain groups. (d) shows the result of the K-means segmentation and (e) the verified points. (f) shows in yellow the best piecewise linear contour when taking the verified points into account.

verified boundary points are the ones on strong edges and with one side classified as the foreground and the other side as the background. The verified boundary points are shown in Figure 3.2e. Notice that most of the boundary points that coincide with the real mask boundary are verified and the ones failed are mostly off the real mask boundary.

3.2.4 Creation of Mask Boundary Candidates

The mask boundary is selected from a set of the mask boundary candidates. The set of candidates are formed in the following two ways: 1) one candidate is created from *verified* boundary points and 2) many candidates are created by grouping the edgel chains. Given a candidate (either verified points or a group of edgel chains), a convex hull is created by using the Graham's scan[72]. Holding the shape unchanged, the convex hull is converted into a piecewise linear contour. The nodes on the convex hull form the control points of the piecewise linear contour. For example, the boundary candidate is shown in Figure 3.3b and the resulting piecewise

linear contour is shown in Figure 3.3c. Each boundary candidate, represented by the convex piecewise linear contour, is to be evaluated in the next section.

Grouping edgel chains is inspired by the observation that the mask boundary is usually detected as edges. But, when mask boundary intersects vessels, the mask boundary is unlikely to be detected in one complete edgel chain, but in multiple segments (Figure 3.3a). To cope with this, we enumerate all combinations of 4 or fewer number of edgel chains (Figure 3.3b. This often results in a large number of candidates. Many of them, however, are unnecessary and slow down the evaluation process. For each combination, we verify that the resulting linear piecewise contour 1) contains the image center and 2) covers at least half of the circle arc with respect to the image center. The latter is to be done in the following steps. 180 rays are emitted from the image center, one for every 2°. The coverage vote of an edgel chain increments when a ray intersects it. During the verification, if the sum of coverage votes of edgel chains is below 90 (half circle as 2° per vote), the combination is discarded. This verification is simple but quite effective — the number of generated candidates is reduced to between a half and one tenth of all possible combinations.

3.2.5 Candidate Evaluation and Selection

Denoting $\mathbf{k}_i$ ($i \in [1...n]$) as the control points on a piecewise linear contour, each linear segment of the contour can be written as:

$$\mathbf{C}_i(u) = \begin{cases} (1-u)\mathbf{k}_i + u\mathbf{k}_{i+1} & \text{if } i \in [1...n-1], \\ (1-u)\mathbf{k}_n + u\mathbf{k}_1 & \text{if } i = n. \end{cases}$$

Though this representation is only continuous but not differentiable, its simplicity and local control property (one control point only affects two neighboring segments) are more desired. More importantly, this piecewise linear model can be used to model an abrupt change on the mask boundary. While other higher-order and smoother contour representations can be used, we will show that this piecewise linear contour is suitable for the masking purpose.

The candidate contours are evaluated with an objective function defined as follows. Denote the set of points on the contour as $\mathcal{C} = \{\mathbf{C}_i(u) | \forall u \in [0,1], \forall i\}$ and

the normal direction of point $\mathbf{p}$ on the contour as $\boldsymbol{\eta}_p$. Again taking the original image as $\mathcal{I}$, we can write the gradient as $\boldsymbol{\mathcal{I}}_g$ and the binary image of verified boundary points as $\mathcal{I}_v$. Then the objective function can be written as:

$$E = -\sum_{\mathbf{p}\in\mathcal{C}} \{\mathcal{I}(\mathbf{p}) + c_1|\boldsymbol{\eta}_p^\top \boldsymbol{\mathcal{I}}_g(\mathbf{p})| + c_2\mathcal{I}_v(\mathbf{p})\} \tag{3.4}$$

The objective function is designed to pull the contour towards pixels with high intensities and high gradients and towards verified points. We have chosen empirically the coefficients to be $c_1 = 30$ and $c_2 = 100$.

In the set of boundary candidates, the one that minimizes this objective function is selected as the mask boundary. In Section 3.3, this mask boundary detected at the coarse level will be refined in a coarse-to-fine manner.

3.3 Refinement

We adopt active contours for refinement of the mask boundary selected by the foregoing method. An active contour (or a "snake"), first introduced by Kass in [100], is a curve or a contour that evolves from its initial position toward the position that minimizes an objective function. The optimization is usually carried out by variational calculus approaches. One of the most commonly used solves the Euler-Lagrange necessary condition [100] for the solution representing the optimum curve/contour. There are a few disadvantages to this approach:

1. There is no guarantee on either the global or the local optimality because it is not a sufficient condition.

2. The use of higher-order derivatives, resulting from the Euler-Lagrange condition, often leads to numerical instability.

3. It is often difficult to incorporate hard constraints.

Similar to active contours, level set methods are also widely used to evolve contours [37, 102, 131, 156, 157, 159, 185]. Compared to active contours, level set methods usually do not preserve topology [6], i.e., contours can split or merge. But

[6] There are topology-preserving level set methods [78]. But they are not of concerns here.

level set methods also suffer from the same disadvantages as above. Moreover, as a retina surface always appears as *one* region in the image, we prefer to maintain the topology of contour. Therefore, we are not going to take the level set approach.

In 1990, Amin proposed the use of dynamic programming to optimize the active contour [3]. Dynamic programming guarantees the global optimality within the search window. It is easy to incorporate hard constraints and different internal energy terms. Also, dynamic programming does not require evaluating any higher order derivatives than the ones presented in the objective function and hence they are less sensitive to noise.

There are disadvantages to using dynamic programming. First, the time complexity of dynamic programming is related to the contour representation, the objective function, and to the hard constraints. — usually exponential in the number of control points needed to evaluate the objective function or the constraints (will be discussed later). Second, the search space is discretized —- only a small number of fixed steps can be taken during each iteration. Nevertheless, the ability to adopt hard constraints and the global convergence property outweigh the disadvantages. Besides, the discrete search space is not a concern because, for the masking purpose, the mask boundary can usually tolerate inaccuracy within a few pixels. The accuracy of masking algorithm will be evaluated in the Section 3.4. The issue of time complexity will be discussed in more detail in Section 3.3.2.

The objective function used in the active contour technique is expanded from Equation 3.4 to include regions inside and outside of the contour. As shown in Figure 3.4, points within two strips along the contour will be considered in the new objective function. We still denote the normal direction of point $\mathbf{p}$ on the contour as $\boldsymbol{\eta}_p$, but now have it pointed to the interior of the region enclosed by the contour. Pixels within the strip located inside the contour can be expressed as $\mathbf{p}+j\boldsymbol{\eta}_p$, $j > 0$, whereas the pixels on the outside strip as $\mathbf{p} - j\boldsymbol{\eta}_p$. The new objective function is

written as:

$$
\begin{aligned}
F &= E + E^{+} + E^{-} \qquad (3.5)\\
E^{+} &= -\sum_{\mathbf{p}\in\mathcal{C}}\sum_{j=0}^{j_0}\left\{w_G(\frac{j}{j_0})\mathcal{I}(\mathbf{p}+j\boldsymbol{\eta}_p) - c_3 Q(\mathcal{I}(\mathbf{p}+j\boldsymbol{\eta}_p))\right\}\\
E^{-} &= -\sum_{\mathbf{p}\in\mathcal{C}}\sum_{j=0}^{j_0}\left\{-w_G(\frac{j}{j_0})\mathcal{I}(\mathbf{p}-j\boldsymbol{\eta}_p) - c_1|\boldsymbol{\eta}_p^T\boldsymbol{\mathcal{I}}_g(\mathbf{p}-j\boldsymbol{\eta}_p)|\right\}\\
Q(x) &= \begin{cases} -\frac{x-\mu_b}{\sigma_b} + a & \text{if } x < \mu_b + a\sigma_b,\\ 0 & \text{otherwise}\end{cases}
\end{aligned}
$$

where E, defined in Equation 3.4, is the cost evaluated along the contour and E^{+} (E^{-}) is the cost evaluated inside (outside) the contour. w_G is a Gaussian weighting function.

The function Q penalizes pixels that have similar appearances as the background. As we discussed before, the background (mask) is generally dark and homogeneous, with the exception of tagging identifiers. Once we determine the initial mask contour, the mean μ_b and the standard deviation σ_b can be estimated robustly from pixels outside of the contour using IRLS method and the formulation presented in Equation 3.1. This estimation is done only once and the statistics are held fixed throughout the refinement process. The constant a is set to 1 and c_3 to 150.

Putting this together, E^{+} maximizes pixel intensities and penalizes pixels similar to the background. By contrast, E^{-} minimizes pixel intensities and penalizes the gradient normal to the contour, which generates a pulling force when there is a parallel edge outside of the contour.

The optimization is carried out by dynamic programming as described in [3]. We rewrite the objective function F (Equation 3.5) with respect to the control points $\mathbf{u}_i$ of linear segments:

$$F(\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_n) = F_1(\mathbf{k}_1, \mathbf{k}_2) + F_2(\mathbf{k}_2, \mathbf{k}_3) + \ldots + F_n(\mathbf{k}_n, \mathbf{k}_1)$$

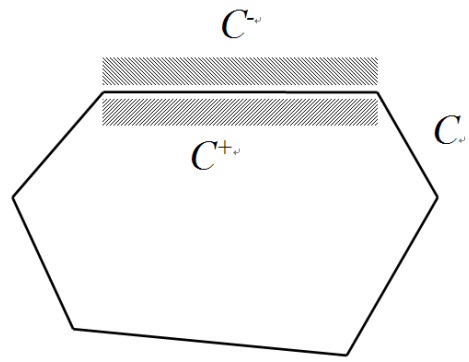


Figure 3.4: This shows the strips used in the objective function: $\mathcal{C}^-$ is the strip outside of the contour (background), and $\mathcal{C}^+$ is the strip inside of the contour (foreground),

We address one important issue not mentioned in Amin's paper [3]. As a closed contour, the fact that the first control point $\mathbf{k}_1$ appears on both the first segment $F_1(\mathbf{k}_1, \mathbf{k}_2)$ and the last $F_n(\mathbf{k}_n, \mathbf{k}_1)$ violates the requirements of dynamic programming. The solution is to apply exhaustive search on the first control point $\mathbf{k}_1$. By holding $\mathbf{k}_1$ fixed at each possible position, we use dynamic programing to solve for the remaining control points. We define a partial sum function as follows:

$$\begin{aligned}
G_2(\mathbf{k}_2) &= F_1(\mathbf{k}_1, \mathbf{k}_2) \\
G_3(\mathbf{k}_3) &= \min_{\mathbf{k}_2} \left[G_2(\mathbf{k}_2) + F_2(\mathbf{k}_2, \mathbf{k}_3)\right] \\
G_4(\mathbf{k}_4) &= \min_{\mathbf{k}_3} \left[G_3(\mathbf{k}_3) + F_3(\mathbf{k}_3, \mathbf{k}_4)\right] \\
&\vdots \\
G_i(\mathbf{k}_i) &= \min_{\mathbf{k}_{i-1}} \left[G_{i-1}(\mathbf{k}_{i-1}) + F_{i-1}(\mathbf{k}_{i-1}, \mathbf{k}_i)\right] \\
&\vdots \\
\hat{G} &= \min_{\mathbf{k}_n} \left[G_n(\mathbf{k}_n) + F_n(\mathbf{k}_n, \mathbf{k}_1)\right]
\end{aligned} \tag{3.6}$$

$\hat{G}$ is the minimum value of the objective function (Equation 3.5). The contour resulting in this minimum value can be found out by the *backward* method: for each control point $i = \{n-1, n-1, \ldots, 2\}$, the optimum location of $\mathbf{k}_i$ is the one that minimizes $G_i(\mathbf{k}_i) + F_i(\mathbf{k}_i, \mathbf{k}_{i+1})$.

We denote the search window size as m ($m = 4$ as we search over 4 immediate neighboring pixels) and still use n to denote the number of control points of the piecewise linear contour. At each step of dynamic programming, there are m entries of $G_i(\mathbf{k}_i)$ values to be filled in. For each entry, the minimization is carried out over m possible values. Therefore, the dynamic programming procedure has complexity $O(nm^2)$. Combining with the exhaustive search for the first control point, the complexity for each iteration of contour evolution is $O(nm^3)$.

3.3.1 Hard Constraints

The convexity constraint, which is difficult to enforce in the variational calculus approaches, now can easily be incorporated into dynamic programming. It can be achieved by enforcing the cross product of directional vectors between consecutive control points to be non-positive:

$$H(\mathbf{k}_{i-1}, \mathbf{k}_i, \mathbf{k}_{i+1}) = (\mathbf{k}_{i+1} - \mathbf{k}_i) \times (\mathbf{k}_i - \mathbf{k}_{i-1}) \leq 0, \tag{3.7}$$

where the 2D cross product is defined as $\mathbf{u} \times \mathbf{v} = u_0 v_1 - u_1 v_0$.

Combining the convexity constraint with the objective function F (Equation 3.5), we obtain a new objective function F':

$$F'(\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_n) = \sum_{i=1}^{n-2} F'_i(\mathbf{k}_i, \mathbf{k}_{i+1}, \mathbf{k}_{i+2}) + F'_{n-1}(\mathbf{k}_{n-1}, \mathbf{k}_n, \mathbf{k}_1) + F'_n(\mathbf{k}_n, \mathbf{k}_1, \mathbf{k}_2).$$

Each segment is

$$F'_i(\mathbf{k}_i, \mathbf{k}_{i+1}, \mathbf{k}_{i+2}) = \begin{cases} F_i(\mathbf{k}_i, \mathbf{k}_{i+1}) & \text{if } H(\mathbf{k}_i, \mathbf{k}_{i+1}, \mathbf{k}_{i+2}) \leq 0, \\ +\infty & \text{if } H(\mathbf{k}_i, \mathbf{k}_{i+1}, \mathbf{k}_{i+2}) > 0 \end{cases} \tag{3.8}$$

Because three control points are needed to evaluate a segment (instead of two),

the new partial sum function G' has two parameters (instead of one):

$$\begin{aligned} G'_2(\mathbf{k}_2, \mathbf{k}_3) &= F'_1(\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3) \\ &\vdots \\ G'_i(\mathbf{k}_i, \mathbf{k}_{i+1}) &= \min_{\mathbf{k}_{i-1}} \left[G'_{i-1}(\mathbf{k}_{i-1}, \mathbf{k}_i) + F'_{i-1}(\mathbf{k}_{i-1}, \mathbf{k}_i, \mathbf{k}_{i+1}) \right] \\ &\vdots \\ \hat{G}' &= \min_{\mathbf{k}_n} \left[G'_n(\mathbf{k}_n, \mathbf{k}_1) + F'_n(\mathbf{k}_n, \mathbf{k}_1, \mathbf{k}_2) \right]. \end{aligned} \tag{3.9}$$

3.3.2 Efficiency of Contour Evolution

As mentioned above, the time complexity increases exponentially w.r.t. the number of parameters needed to evaluate the objective function or the constraints. Now that three control points are needed to evaluate a segment $F'_i(\mathbf{k}_i, \mathbf{k}_{i+1}, \mathbf{k}_{i+2})$, the time complexity increases to $O(nm^5)$, because of m^2 entries of $G'_i(\mathbf{k}_i, \mathbf{k}_{i+1})$ and m^2 entries for the exhaustive search.

Notice the evaluation of $F_i(\mathbf{k}_i, \mathbf{k}_{i+1})$ is the most expensive among all operations because the computation involves examining a large number of pixels within the forementioned strips (Figure 3.4) from $\mathbf{k}_i$ to $\mathbf{k}_{i+1}$. A straight-forward implementation of dynamic programming using Equation 3.9 needs to evaluate $F_i(\mathbf{k}_i, \mathbf{k}_{i+1})$ $O(nm^5)$ times, which is expensive.

The solution is in the structure of $F'_i(\mathbf{k}_i, \mathbf{k}_{i+1}, \mathbf{k}_{i+2})$ shown in Equation 3.8. Notice only two parameters $\mathbf{k}_i$ and $\mathbf{k}_{i+1}$ are needed to evaluate $F_i(\mathbf{k}_i, \mathbf{k}_{i+1})$. We create a table to cache all values of $F_i(\mathbf{k}_i, \mathbf{k}_{i+1})$ for each iteration of contour evolution. As there are n nodes in the piecewise linear contour, the table has n rows and m^2 columns, which requires nm^2 evaluations of $F_i(\mathbf{k}_i, \mathbf{k}_{i+1})$ to fill in. Other operations, such as the cross product $H(\mathbf{k}_{i-1}, \mathbf{k}_i, \mathbf{k}_{i+1})$, are trivial to compute. Through the use of this cache table, the running time is now reduced close to when using the objective function F in Equation 3.5 (without the convexity constraint).

This solution implies the complexity of active contours solved by dynamic programming is closely related to the contour representation. In comparison, the addition of hard constraints or internal force terms, such as the thin plate energy, has only small effect on the computation time (when taking the above approach). This

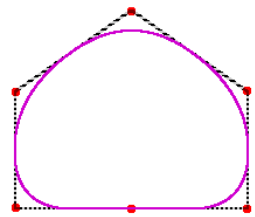


Figure 3.5: Curve smoothing in Bézier form. Piecewise linear contour is shown in Dotted lines, with control points in red points. The resulting smooth curve is shown in magenta.

observation justifies our choice on piecewise linear contour representation as this representation, though not smooth, results in low time complexity when evolving using dynamic programming. One may argue that piecewise linear contour is not smooth and unfit for human aesthetics. In the following section, we will show that, with a simple additional modification, the resulting contour is smooth.

3.3.3 Contour Smoothing and Mask Image Generation

The refinement is completed once it reaches the finest scale. As a final step, the resulting piecewise linear contour is replaced by a quadratic spline in Bézier form[55, Ch.3,7] for better visualization. The Bézier form has the advantage that the estimated curve resides within the convex hull (shown in Figure 3.5).

In general, for a control point $\mathbf{k}_{i+1}$ and the two neighboring control points $\mathbf{k}_i$ and $\mathbf{k}_{i+2}$ on the piecewise linear contour, the corresponding Bézier spline segment is given by

$$\mathbf{B}_i^2(t) = (1-t)(\mathbf{k}_i + \mathbf{k}_{i+1})/2 + 2t(1-t)\mathbf{k}_i + t^2(\mathbf{k}_{i+1} + \mathbf{k}_{i+2})/2$$

It can easily be shown that the Bézier spline is continuous and first-order differentiable.

As can be seen from Figure 3.5, the Bézier spline is tangential at the midpoint of each linear segment (when $t = 0$ and $t = 1$) and is smooth around the sharp corners. It will be shown in the Section 3.4 that the mask boundary computed using the Bézier spline form is very close to the ground-truth mask image.

At the end of the computation, a binary image is generated as the final result: pixels inside the mask boundary contour (the retina surface) are labeled with "1", whereas pixels outside (the mask) are labeled with "0". This binary label image is also called "mask image" for short.

3.4 Experiments

The experiments are conducted on four sets of retinal images.

- "Development" set contains 29 greyscale and color images. These images are used in the development of the masking method.
- "Digital" set contains 380 greyscale images, each with resolution 1024×1024. These images were acquired from a single digital fundus camera and share one common mask.
- "Slides" set contains 1182 color images with resolution 1472×1000 scanned from positive slides. These slides had been taken from eyes with various pathologies, such as those caused by Age-related Macular Degeneration and diabetic retinopathy, with acquisition time spanning more than 10 years. The images have masks similar in shape, but shifted and rotated during the scanning process. There are also artifacts from scanning process or from the acquisition, such as tagging identifiers, dirts and hairs.
- "Misc" set contains 546 greyscale and color images with resolutions ranged from 500×430 to 4008×2672, including scanned slides, digital images, fluorescein angiogram sequences. The masks of these images vary in shape, in size, in location and in orientation.

The parameters (mainly the coefficients used in the objective function in Equation 3.5) are chosen based on the observations made with the "development" set. Then, the masking method, with parameters fixed, is applied blindly to the other three sets. Taking one retinal image as the input, the masking method described as above produces a binary-label mask image (described in Section 3.3.3) as the

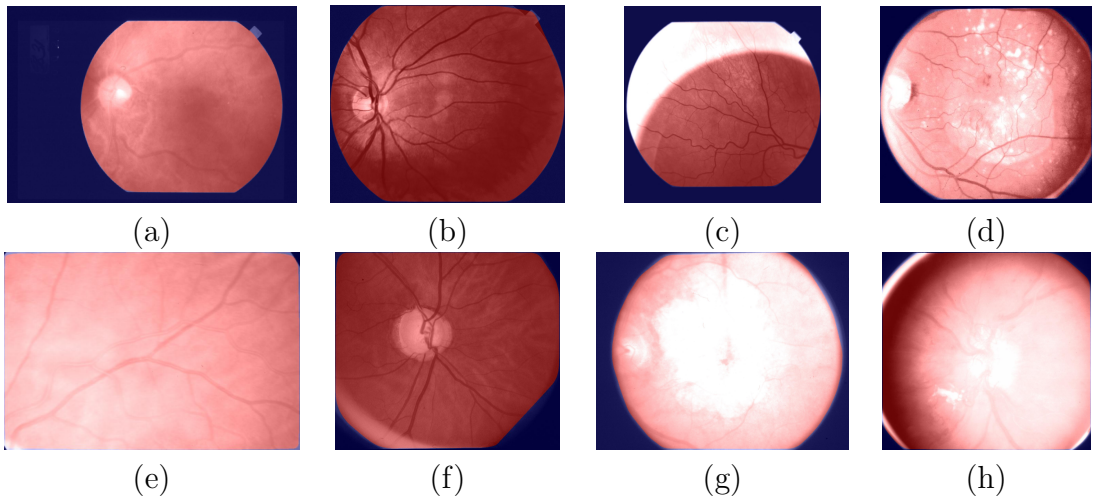


Figure 3.6: The masking results of images in Figure 1.6. Masks are superimposed on the images. Red indicates the detected retina and blue indicates the detected mask.

output. As an illustration, the mask images superimposed on the original for the images shown in Figure 1.6 are demonstrated in Figure 3.6.

The first test is a quantitative experiment on the quality of the generated mask. The experiment is conducted on the "digital" set. Since all the images in this set share a common mask, we can compare the automatically generated mask images with the ground truth. The ground-truth mask image is extracted manually in a photo-editing software (GIMP) using a simple thresholding operation as the background is homogeneous with a fixed intensity value. When comparing the generated mask image with the ground truth, the number of pixels with incorrect labels are counted. The summary of the comparison is presented in Table 3.1. Overall, fewer than 1% of the pixels are mis-classified for any of the 380 images.

For the other two sets — "slides" and "misc", the ground truth is not available. Because the masks vary across images, it is too difficult to create the ground-truth mask image for each individual image. Therefore, the results are evaluated in a qualitative way: for each image, the mask image is superimposed on the original, which is visually examined by a fellow graduate student. A mask image is deemed as correct if 1) the generated mask boundary is close to the real boundary and 2) it does not mislabel the background area (but it can mislabel a few foreground pixels

	false retina		false mask	
	pixels	percentage	pixels	percentage
Min	10	0.00%	1821	0.17%
max	5446	0.51%	10294	0.98%
Average	831	0.08%	4235	0.40%

Table 3.1: Quantitative results on the quality of masking using the 380-image "digital" set. The generated mask images are compared with the manually-extracted ground-truth mask image. The column marked as "false retina" indicates the number of pixels labeled as the retina but in fact are the mask, whereas "false mask" indicates the number of pixels labeled as mask but in fact are the retina.

around the mask boundary as those pixels are not very useful due to vignetting). The number of correct and incorrect masking is summarized in Table 3.2. These experiments were conducted on a Pentium IV 3.2GHz PC. The execution time varies from 2s to 15s.

In Section 3.2, during the detection of mask, we combine edge with intensity information to guide the generation of mask boundary candidates. To examine why both are needed, we conduct the following experiments on the 29-image "development" set. First, we apply the complete masking technique, resulting in 28 success and 1 failure (also determined by the fellow student). Second, when taking out the modified K-means, which results in no verified points, the number of failures increases to 4. Third, when taking out the Canny edge detection, which results in only one mask boundary candidate —the verified points, the number of failures increases to 16. The increases on the number of failures indicates that it is necessary to combine edge and intensity information and form multiple boundary candidates.

In Section 3.3.2, we propose a cache table to speed up the active contour optimization using dynamic programming. To extract all 29 mask images from the "development" set, our automatic method took 150 seconds in total, whereas without the cache table, it took 555 seconds in total. The speedup is significant with the cache table.

Figure 3.7 show failure examples of masking. The reasons for failures include 1) a bright region in the mask area (Figure 3.7a) and 2) the background statistics adversely affected by the double boundaries (Figure 3.7b and Figure 3.7c). Ad-

Set Name	total	correct	incorrect
development	29	28	1
digital	380	380	0
slides	1182	1176	6
misc	546	540	6

Table 3.2: Summary of masking results on thress sets of retinal images. "Digital" is the set of images acquired by one digital camera. "Slides" is the set of pathological images that were scanned from slides. "misc" is the set has a variety of images, including digital ones, scaned slides, cropped ones and FA sequence.

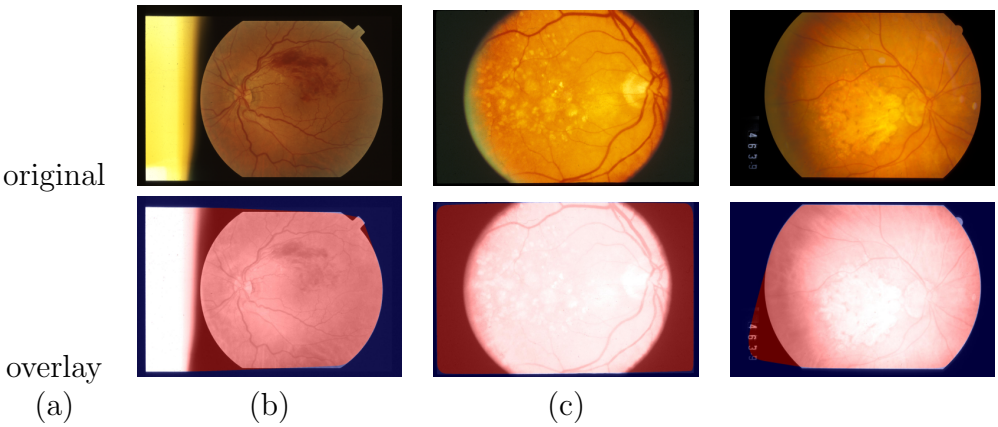


Figure 3.7: Examples of failures. The first row is the original images the second row is images with mask superimposed. The red indicates the foreground and blue indicates the background.

justing parameters according to these images may be able to improve. But unless we establish a model explicitly for images with double boundaries, it is difficult to eliminate all failures in this category.

3.5 Summary

We have presented a fully automatic technique that segments a retina image into the retina region and the mask region. This technique is robust to different camera models, illumination changes, and scanning artifacts and therefore is applicable to a large variety of retina images, including scanned slides, cropped images, digitally-acquired ones, and FA sequences.

Other than the experiments presented in this chapter, the automatic masking

method is also applied as a preprocessing step for image registration algorithms [199, 223] and change detection algorithms [153, 172]. The feedback on its use has been positive thus far.

CHAPTER 4
Covariance-Driven Mosaic Formation from Sparsely-Overlapping Image Sets

4.1 Overview

In this chapter we consider the problem of refining the multi-image alignment of retinal images. This occurs as a later stage of RIVERS after masking (Chapter 3), pairwise image registration [199], and linear joint registration [31]. This problem occurs because in some situations, especially for pairs with small overlap, there are still misalignments in the mosaic. These are due to the failure of pairwise registration and hence the lack of constraints (see Figure 1.8 for example). It is difficult or even impossible to improve the pairwise registration algorithm to align these images, because with such small amount of overlap, there may not even be enough constraints [7] available to produce a stable transformation!

Our approach is to search for new correspondences between these images in the context of the joint registration (mosaicing). These constraints are then incorporated back into multi-image without generating an intermediate pairwise alignment. Though these newly-added constraints are not enough to estimate a stable pairwise estimate, they are enough to bring the two images into alignment in the joint registration, because the joint registration "borrow" constraints from other pairwise alignments. There are two advantages to this approach: a) the overlapping pairs that fail pairwise registration are roughly aligned in the joint registration due to the propagation of the constraints between images and b) the covariance matrix computed during the joint registration can be used to guide the search of new correspondences.

[7] By saying "constraints", we mean feature correspondences that contribute in the estimation of transformation.

4.1.1 Related Literature

The mosaicing problem has received a significant amount of attention both in the computer vision literature [160, 180, 191, 23] and the ophthalmology literature [49, 231]. Several papers focus on mosaicing video sequences, where the overlap between images tends to be relatively high [90, 94, 109, 191]. Sawhney *et al.* addressed the question of low image overlap, using the global inferencing of topological relationship between images to generate additional inter-image constraints [179].

Bundle adjustment technique is well-known in the area of multi-image registration. It simultaneously (or iteratively) refines the transformation estimates and the feature location estimates [220]. This technique is most widely used in structure-from-motion, where the feature locations in the scene must also be estimated [91, 114, 171, 203, 242]. When constructing mosaics, however, we are interested in only the transformations. Thus, several methods on mosaic construction apply a bundle-adjustment-like optimization to enforce consistency between all images, but do not estimate the true feature locations [23, 31, 90, 180, 181, 191]. As the goal of bundle adjustment is to enforce consistency between all images and distribute errors evenly across all feature correspondences, we will loosely refer to the optimization techniques used in the above methods (both structure-from-motion and mosaic construction) as bundle adjustment techniques without making distinctions between them.

Many of these mosaic construction methods [160, 180, 191, 23], including the structure-from-motion methods [23, 31, 90, 180, 181, 191], apply bundle adjustment techniques based on *pre-determined* pairwise relationship. None of them consider or try to fix misalignments occured on image pairs which the pairwise registration method fails upon.

4.1.2 Approach

The major steps of mosaic construction include registering pairs of image, extracting constraints from the registered pairs, and jointly estimating the transformations aligning all images using the extracted constraints. (A final step of constructing the actual mosaic by transforming and blending the images is not discussed

here.) Let $\{\mathcal{I}_0, \ldots, \mathcal{I}_N\}$ be a set of images, and let A be an algorithm for registering pairs of images. We assume $A(\mathcal{I}_i, \mathcal{I}_j)$ produces two types of output if it successfully aligns $\mathcal{I}_i$ to $\mathcal{I}_j$: $\hat{\mathbf{\Theta}}_{i,j}$ is the set of estimated transformation parameter values, and $\mathcal{C}_{i,j}$ is a set of inter-image constraints — corresponding points. For simplicity in this discussion, we assume A is applied to all possible image pairs. The result can be viewed as a connectivity / topology graph: each image forms a vertex, and an edge is formed between vertices i and j when $A(\mathcal{I}_i, \mathcal{I}_j)$ is successful (Figure 1.8). The final step, which we will call the "joint alignment" step, simultaneously estimates the final transformations [191] using all the constraint sets $\mathcal{C}_{i,j}$, essential the bundle adjustment technique [220].

This technique will produce a final set of transformations for all images if the topology graph is connected. This does not ensure accurate alignment, however, especially when the graph is sparsely connected and there is relatively little overlap between some images. Two images, such as $\mathcal{I}_4$ and $\mathcal{I}_5$ of Figure 1.8, may overlap without there being enough information for pairwise registration to succeed. In the context of multi-image registration, constraints from nearby images may not be enough to force consistency between these images during joint alignment. The result can be clear misalignments.[8]

There are three key criteria for the success of a pairwise registration method: 1) the initialization, 2) the number of constraints (correspondences), and 3) the stability of the resulting transformation. Taking $\mathcal{I}_4$ and $\mathcal{I}_5$ in Figure 1.8 as an instance, though there exists a sufficient number of constraints (points sampled along the vessel contour), it is difficult to produce a correct initialization. Second, even if there is an initialization, the final pairwise transformation by itself is not stable — not enough constraints in the vertical direction. When given only $\mathcal{I}_4$ and $\mathcal{I}_5$, Hence, the two images cannot be aligned because of the aforementioned reasons. When given *multiple* images, $\mathcal{I}_1$ to $\mathcal{I}_5$, however, we are able to establish correspondences between $\mathcal{I}_4$ and $\mathcal{I}_5$ and compute a transformation estimate between them.

[8] We stress that the problem is **not** with pairwise registration. There aren't sufficient constraints in the images shown in Figure 1.8 for estimation of the 12-parameter transformation necessary to align retinal images.

The key problem addressed here is how to generate additional constraints, beyond the results of pairwise registration, in order to avoid misalignments in the final, joint alignment. Sometimes the misalignment may be small, which would allow a technique like incremental block matching [191] — a search using a square window of 32, 16, or 8 pixels within the range of half window width — to succeed. At other times, the misalignment may be much greater. We handle this by using the alignment uncertainty to drive the generation of additional constraints. The covariance matrices of the transformation parameters estimated in the joint alignment are computed — even for transformations between image pairs that did not register pairwise. These are used to compute error covariances on the inter-image mapping of feature points, which guides the search for new correspondences. These correspondences are used in turn to refine the joint estimate. The effect of these techniques is illustrated in Figure 4.1. In short, we are using transformation parameter covariances to drive the generation of constraints between images even when pairwise registration fails. These additional constraints are not enough to make pairwise registration succeed, but they are enough to improve the results of joint alignment and eliminate most misalignments in the final mosaic.

In this chapter, the pairwise registration algorithm A is treated as a black box. The actual algorithm [199] works automatically, either producing an accurate alignment between image pairs, or indicating that they can not be aligned. Experimental validation on an extensive data set has shown in [223] that this pairwise registration algorithm does *NOT* very likely produce incorrect alignments.

In the retina application, the alignment of two-dimensional images uses a 12-parameter quadratic transformation model [32, 199]. Given two images, $\mathcal{I}_m$ and $\mathcal{I}_n$, let $\mathbf{p} = (x, y)^T$ be a pixel location in $\mathcal{I}_m$, and $\mathbf{p}' = (x', y')^T$ be the transformed location in $\mathcal{I}_n$. Define $\mathbf{X}(\mathbf{p}) = (x^2, xy, y^2, x, y, 1)^T$. Then the transformation of $\mathbf{p}$ onto $\mathbf{p}'$ is

$$\mathbf{p}' = \mathbf{T}(\mathbf{p}; \mathbf{\Theta}_{m,n}) = \mathbf{\Theta}_{m,n}\mathbf{X}(\mathbf{p}), \tag{4.1}$$

where $\mathbf{\Theta}_{m,n}$ is a 2×6 parameter matrix.[9] This model is accurate to less than a pixel

[9] Abusing notation for the sake of convenience, in other contexts we may interpret $\mathbf{\Theta}_{m,n}$ as a 12×1 parameter vector as well.

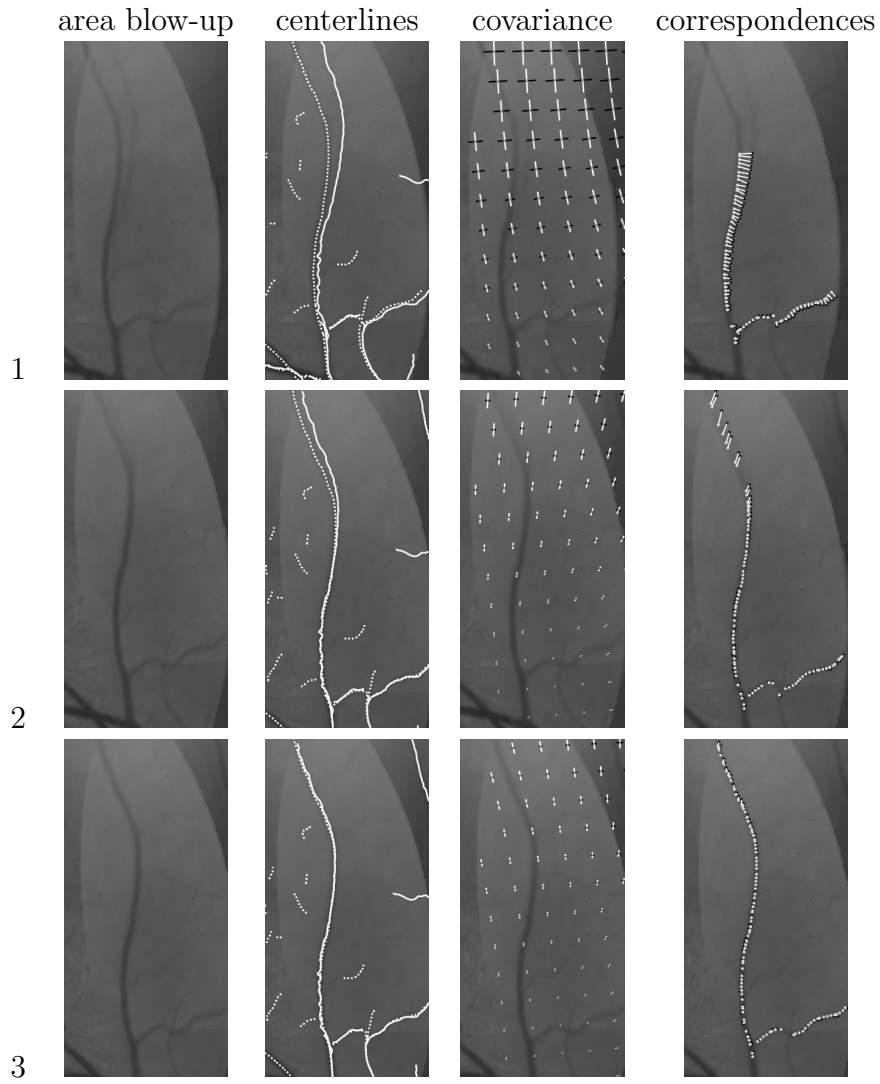


Figure 4.1: Three iterations of the covariance-driven process of adding constraints and refining the transformation estimates, shown for the rectangular region highlighted in Figure 1.8. The 2nd column shows the centerline points of $\mathcal{I}_5$ (solid segments) mapped onto $\mathcal{I}_4$ and its centerline points (dotted segments). The misalignment at the top of the region is most prominent, especially in the first iteration. The 3rd column shows sampled transfer error covariance matrices — the two orthogonal eigenvectors are represented by a cross, with the length proportional to the corresponding eigenvalues. The axis corresponding to the larger eigenvalue is drawn in white. The 4th column shows new generated correspondences. Clearly, as the iterations proceed, the alignment, the covariance matrices and the matching error improve dramatically.

on 1024×1024 retinal images [199].

The pairwise algorithm generates constraints based on the location of blood vessel landmarks — branching and cross-over points of the retinal vasculature — and, more importantly, points along the centerlines of the vasculature [32]. Vessels are used because they are prominent, static and easily detectable. (The appearance of the non-vascular background tends to vary with illumination, and background structures such as waste deposits and pathologies tend to change over time.) The centerline points are discrete samples along fairly straight contours, so that precisely matching and aligning them is unrealistic (the aperture problem for registration). Instead, the constraints for registration are alignment of a centerline point from one image with a linear approximation to the centerline contour in the other. Algebraically, if $\mathbf{p}_{m,i}$ is a centerline point location in image $\mathcal{I}_m$ and if $\mathbf{p}_{n,j}$ is a centerline point location in image $\mathcal{I}_n$ with local contour normal $\hat{\boldsymbol{\eta}}_{n,j}$, then the "normal distance" error of the transformation is

$$(\boldsymbol{\Theta}_{m,n}\mathbf{X}(\mathbf{p}_{m,i}) - \mathbf{p}_{n,j})^T \hat{\boldsymbol{\eta}}_{n,j}. \tag{4.2}$$

This measures the signed distance of the mapping of $\mathbf{p}_{m,i}$ to the line through $\mathbf{p}_{n,j}$ with normal $\hat{\boldsymbol{\eta}}_{n,j}$.

Based on this discussion, we can now define the correspondence sets that emerge from pairwise registration. The set of image pairs for which A succeeds is $\mathcal{P} = \{(m,n)\}$. Here, order matters because pairwise registration may in rare cases succeed from $\mathcal{I}_m$ to $\mathcal{I}_n$, but not vice-versa. For each $(m,n) \in \mathcal{P}$, the constraint set is

$$\mathcal{C}_{m,n} = \{(\mathbf{p}_{m,i}, \hat{\boldsymbol{\eta}}_{m,i}; \mathbf{p}_{n,j}, \hat{\boldsymbol{\eta}}_{n,j})\}. \tag{4.3}$$

Finally, each constraint in each constraint set has a weight associated with it, $w_{m,n;i,j}$, computed using robust M-estimator weighting [200]. It is important to note that this weight is the usual robust weight multiplied by $1/\sigma^2_{m,n}$, where $\sigma^2_{m,n}$ is the robustly computed variance of the pairwise alignment error between $\mathcal{I}_m$ and $\mathcal{I}_n$. This normalizes the weights appropriately for different image pairs.

4.2 Joint Alignment

The first consideration in developing our technique is to present the joint alignment estimation equations. By specially designating one image $\mathcal{I}_0$ as the "anchor image" on which to construct the mosaic (e.g. $\mathcal{I}_4$ in Figure 1.8), our goal is to estimate the transformations $\mathbf{\Theta}_{1,0}, \ldots, \mathbf{\Theta}_{N-1,0}$ of the remaining $N-1$ images onto this anchor using the set of constraint sets $\{\mathcal{C}_{m,n}\}$. This generates $N-1$ of the $N(N-1)$ interimage transformations. We can choose each of the other images as the anchor in turn in order to estimate the remaining interimage transformations. This capability is important to the constraint generation procedure described below.

For a designated anchor, we divide the set of image pairs $\mathcal{P}$ in two: $\mathcal{P}_D = \{(m,0) \mid (m,0) \in \mathcal{P}\}$ and $\mathcal{P}_I = \{(m,n) \mid m \neq 0, n \neq 0, (m,n) \in \mathcal{P}\}$. In words, $\mathcal{P}_D$ is the set of pairs involving the anchor directly ("direct constraints"), while $\mathcal{P}_I$ is the set of pairs that do not involve the anchor image ("indirect constraints"). These two constraint sets are treated separately in the joint alignment.

We have two choices in computing the joint alignment. The first, used in our previous work in matching vascular landmarks [31], ignores the normal directions in order to define a least-squares error norm based on Euclidean feature-point distances:

$$\begin{aligned} E(\mathbf{\Theta}_{1,0}, \ldots, \mathbf{\Theta}_{N,0}) = & \\ & \sum_{(m,0)\in\mathcal{P}_D} \sum_{(i,j)\in\mathcal{C}_{m,0}} w_{m,0;i,j} \|\mathbf{\Theta}_{m,0}\mathbf{X}(\mathbf{p}_{m,i}) - \mathbf{p}_{0,j}\|^2 \\ + & \sum_{(m,n)\in\mathcal{P}_I} \sum_{(i,j)\in\mathcal{C}_{m,n}} w_{m,n;i,j} \|\mathbf{\Theta}_{m,0}\mathbf{X}(\mathbf{p}_{m,i}) - \mathbf{\Theta}_{n,0}\mathbf{X}(\mathbf{p}_{n,j})\|^2. \end{aligned} \tag{4.4}$$

The first term measures errors against feature locations in the anchor image, while the second term measures inconsistencies in the mapping of corresponding, but non-anchor features. This is quadratic in the transformation parameters and therefore can be solved non-iteratively.

The second choice, introduced here, defines a weighted least-squares error norm

based on normal distances:

$$
\begin{aligned}
&E_{\hat{\boldsymbol{\eta}}}(\boldsymbol{\Theta}_{1,0}, \ldots, \boldsymbol{\Theta}_{N,0}) = \\
&\quad \sum_{(m,0)\in\mathcal{P}_D} \sum_{(i,j)\in\mathcal{C}_{m,0}} w_{m,0;i,j}[(\boldsymbol{\Theta}_{m,0}\mathbf{X}(\mathbf{p}_{m,i}) - \mathbf{p}_{0,j})^T \hat{\boldsymbol{\eta}}_{0,j}]^2 \\
&\quad + \sum_{(m,n)\in\mathcal{P}_I} \sum_{(i,j)\in\mathcal{C}_{m,n}} w_{m,n;i,j}[(\boldsymbol{\Theta}_{m,0}\mathbf{X}(\mathbf{p}_{m,i}) - \boldsymbol{\Theta}_{n,0}\mathbf{X}(\mathbf{p}_{n,j}))^T \hat{\boldsymbol{\eta}}'_{n,j}]^2. \qquad (4.5)
\end{aligned}
$$

The issue here is that the transformation of each centerline point normal direction, $\hat{\boldsymbol{\eta}}'_{n,j}$, depends on the transformation parameters. Thus, we no longer have a quadratic estimation problem and must resort to an iterative technique. We initialize the parameters using Euclidean distances as in (4.4). Then we alternate (a) estimating the normals using the Jacobian of the transformations and (b) re-computing the estimates using (4.5) and fixed normals. This converges quickly.

The algorithm in Can *et al.* [31] using the objective function (4.4) stops at this point. The new technique described in this chapter then generates covariance matrices of all transformation parameter estimates and uses them to guide the generation of new constraints.

4.2.1 Covariance Matrices

The last step of our joint alignment estimation algorithm for a fixed set of constraints is to compute the covariance matrix of the estimated parameters. We can obtain an approximate covariance by inverting the Hessian matrix of (4.5) evaluated at the estimate [169, Ch. 15] [154, Ch. 7]:

$$
\boldsymbol{\Sigma}_{\boldsymbol{\Theta}} = \mathbf{H}^{-1}(E_{\hat{\boldsymbol{\eta}}}(\hat{\boldsymbol{\Theta}})). \qquad (4.6)
$$

Normally, this would be multiplied by the variance of the alignment errors, but these values are already factored into the robust weights used in (4.5). The result is a $12N \times 12N$ matrix. It can be shown that the individual covariance matrices $\boldsymbol{\Sigma}_{i,0}$ are obtained by simply extracting the appropriate 12×12 subblock.

Obtaining this covariance matrix is the major reason why the more complicated objective function in (4.5) is used instead of (4.4): it gives a more reliable

indication of the uncertainty. To demonstrate this intuitively, consider for example the alignment of two images with two parallel lines each, and a fixed set of correspondences along these lines (Figure 4.2). Using the Euclidean distance in (4.4), a shift of the transformation along the lines (Figure 4.2d and Figure 4.2e) would increase the error proportionally, even though the lines themselves would still be registered. Thus, the transformation would appear much more stable that it truly is. Using (4.5) would correctly make this transformation appear unstable. This is crucial for correctly guiding the matching process.

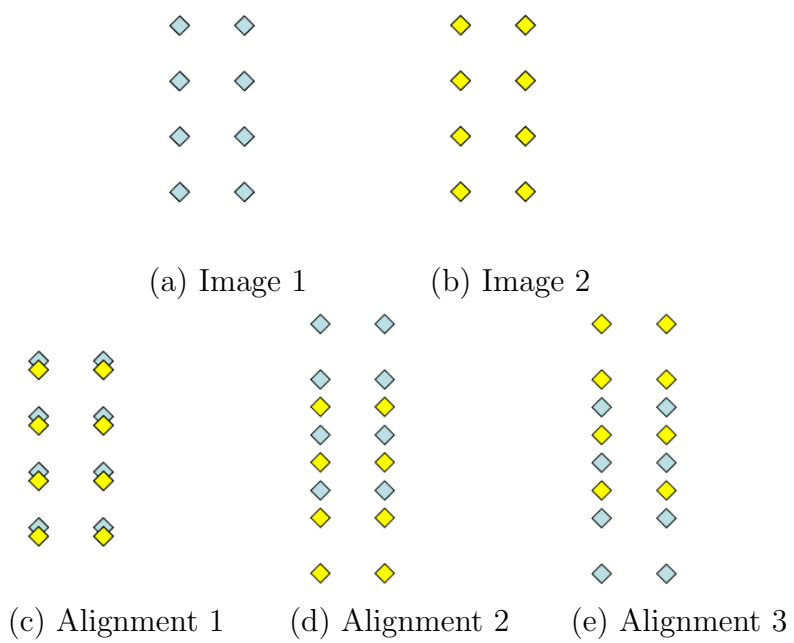


Figure 4.2: Demonstration of uncertainty in transformation when aligning two lines. (a) and (b) show points sampled on two lines. (c) shows an alignment between Image 1 and Image 2, while (d) and (e) show alignments where Image 1 (or (a)) slides up and down along the lines. The covariance matrix computed from Equation 4.5 correctly indicates the uncertainty whereas the one from Equation 4.4 does not.

.

4.3 Generating New Constraints

Our next step, and the most important innovation of the chapter, is to use the joint alignment transformation estimates and covariance matrices to generate

new correspondences. In doing so, we consider any pair of images $\mathcal{I}_m$ and $\mathcal{I}_n$ that overlap (based on the joint alignment) or nearly overlap and that were not aligned successfully by the pairwise registration algorithm. These image pairs generally have low overlap — in practice overlap for these pairs is always lower than 35% of the image area and is generally much lower. We do not consider adding constraints for the image pairs that were aligned by pairwise registration because the pairwise constraints are always sufficient for accurate results in the joint alignment step.

Once the unregistered pairs are identified, each pair is tested in turn for the generation of new constraints. To simplify the process, pair (m, n) is only tested when $\mathcal{I}_n$ is the anchor image, i.e. $n = 0$. All pairs are eventually considered by varying the choice of anchor image. It is important to note that we obtain a covariance matrix for the mapping of $\mathcal{I}_m$ onto $\mathcal{I}_0$ even though there is no pairwise result. This covariance depends on indirect constraints (see Equation 4.5).

We infer new constraints relative to the anchor, iteratively switching which image is chosen as the anchor, but always using all constraints in refinement. For pair $(m, 0)$, the following steps are repeated until the estimate converges

1. Initialize an empty constraint set $\mathcal{C}_{m,0}$.

2. Identify the centerline points in both images that fall in or near the apparent overlap region based on the transformation estimate $\hat{\mathbf{\Theta}}_{m,0}$. Denote these as $\mathcal{P}_m$ and $\mathcal{P}_0$, respectively (suppressing the dependence of each set on the other image in the notation).

3. For each centerline point $\mathbf{p}_{m,i} \in \mathcal{P}_m$:

   (a) Map $\mathbf{p}_{m,i}$ onto $\mathcal{I}_0$ to compute location

   $$\hat{\mathbf{p}}'_{m,i} = \hat{\mathbf{\Theta}}_{m,0}\mathbf{X}(\mathbf{p}_{m,i})$$

   (b) Compute the uncertainty covariance matrix of this mapping $\mathbf{\Sigma}_{\hat{\mathbf{p}}'_{m,i}}$ (Figure 4.1).

   (c) Find the centerline point $\mathbf{p}_{0,j}$ from $\mathcal{I}_0$ minimizing the square Mahalanobis

distance to this transformed location. In particular,

$$\mathbf{p}_{0,j} = \underset{\mathbf{p}\in\mathcal{P}_0}{\operatorname{argmin}}(\hat{\mathbf{p}}'_{m,i} - \mathbf{p})^T \boldsymbol{\Sigma}^{-1}_{\hat{\mathbf{p}}'_{m,i}} (\hat{\mathbf{p}}'_{m,i} - \mathbf{p}) \tag{4.7}$$

(d) Add the constraint $(\mathbf{p}_{m,i}, \hat{\boldsymbol{\eta}}_{m,i}; \mathbf{p}_{0,j}, \hat{\boldsymbol{\eta}}_{0,j})$ to the constraint set $\mathcal{C}_{m,0}$ (Figure 4.1).

4. Estimate the transformation error scale $\sigma_{m,0}$ and compute the weights $w_{m,i;0;j}$ for the new correspondence set.

5. Temporarily add the image pair $(m, 0)$ to the set of direct constraint pairs, $\mathcal{P}_D$, and re-estimate the joint alignment transformations and covariances as described in Section 4.2 using all direct and indirect constraint sets.

After convergence, a test is made to verify the consistency of the final constraint set $\mathcal{C}_{m,0}$. If it passes, image pair $(m, 0)$ is added to the set of image pairs, $\mathcal{P}$, and $\mathcal{C}_{m,0}$ is retained. Thus, constraints between images $\mathcal{I}_m$ and $\mathcal{I}_0$ are added, even though pairwise registration did not succeed.

The rest of this section describes a few of the steps in more detail.

4.3.1 Mapping Error and Matching

The uncertainty in point location mapping is computed from the covariance of the transformation parameter estimate. This uses the forward transfer error [82, Ch. 4]. The mapped point $\hat{\mathbf{p}}'_{m,i} = \hat{\boldsymbol{\Theta}}_{m,i}\mathbf{X}(\mathbf{p}_{m,i})$ is a random variable because it relies on the transformation, which is also a random variable [10]. Its covariance matrix can be approximated from $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\Theta}}_{m,0}}$ and the Jacobian, $\mathbf{J}$, of the transformation evaluated at $\mathbf{p}_{m,i}$:

$$\boldsymbol{\Sigma}_{\hat{\mathbf{p}}'_{m,i}} = \mathbf{J}\boldsymbol{\Sigma}_{\hat{\boldsymbol{\Theta}}_{m,0}}\mathbf{J}^T \tag{4.8}$$

Examples of these transfer error covariance matrices are shown in Figure 4.1. Notice how these change spatially and, as expected, have their major uncertainty axis along the vessel directions.

[10] For simplicity, we do not treat $\mathbf{p}_{m,i}$ as a random variable because feature location error is generally much small than errors in the transformation.

When searching for the correspondence, minimizing the Mahalanobis distance for mapped point $\hat{\mathbf{p}}'_{m,i}$ is more challenging than minimizing Euclidean distance. This minimization is accomplished in two steps. First, with the centerline points organized into a spatial data structure, our algorithm first gathers all possible $\mathcal{I}_0$ centerline points in a square region surrounding $\hat{\mathbf{p}}'_{m,i}$ whose width is determined by the maximum eigenvalue of $\mathbf{\Sigma}_{\hat{\mathbf{p}}'_{m,i}}$. Second, the correspondence is found by exhaustively searching this (generally small) set for the centerline point minimizing the Mahalanobis distance.

4.3.2 Scale Estimation and Weight Calculation

Denote by $r^2_{m,0;i,j}$ the square Mahalanobis distance between the transformed point $\hat{\mathbf{p}}'_{m,i} = \hat{\mathbf{\Theta}}_{m,0}\mathbf{X}(\mathbf{p}_{m,i})$ and its matching point $\mathbf{p}_{0,j}$. Prior to re-estimating the joint alignment, we use the set of these distances to robustly estimate scale and to weight each individual correspondence. The scale, $\sigma_{m,0}$, is computed using a technique that automatically estimates and adjusts for the fraction of inliers [148]. The weights are computed using standard M-estimator weighting, multiplied as above by the inverse variance:

$$w_{m,0;i,j} = \frac{1}{\sigma^2_{m,0}} \frac{\rho'(r_{m,0;i,j}/\sigma_{m,0})}{r_{m,0;i,j}/\sigma_{m,0}}. \tag{4.9}$$

The Beaton-Tukey robust loss function is as follows [10]:

$$\rho(u) = \begin{cases} \frac{a^2}{6}[1-(1-(\frac{u}{a})^2)^3], & |u| \leq a \\ \frac{a^2}{6}, & |u| > a \end{cases} \tag{4.10}$$

4.3.3 Verification of Constraint Sets

After the iterations of match generation, weight calculation, and transformation re-estimation have converged, the constraint set must still be validated. This decides if image pair $(m, 0)$ will be added to the constraint set. The test is simple. For each generated match, let $d_{m,i;0,j}$ be the normal distance, computed as in

Equation 4.2. Compute the weighted average of these distances:

$$e_{m,0} = \sum w_{m,0;i,j} |d_{m,0;i,j}| \ / \ \sum w_{m,0;i,j}.$$

We call this measure the "Centerline Error Measure" or CEM [32, 199]. The constraint set is verified and retained permanently if $e_{m,0}$ is lower than a threshold, empirically determined in previous work to be 1.5 pixels. CEM, with the same threshold, is used to verify pairwise matching results. Thus, covariances and the Mahalanobis distance are used to "pull in" new correspondences for unregistered image pairs, but the final verification is as stringent as in pairwise registration.

4.4 Experiments

We have applied this technique to image sets taken from a large patient database of retinal images. We selected a few to present here to illustrate the algorithm. In some cases, we've chosen a subset of the data sets to obtain a sparser set of images. In other cases, such as shown in Figure 4.3, the original image set has sparse overlap on the periphery of the retina. (Here we only show images from above and to the right of the optic disk.) These are representative of the images being acquired with newer imaging systems.

We can evaluate the results in several ways. The first is a simple visual evaluation of the mosaics constructed with and without the application of the new technique. This is perhaps the most important, but also the most subjective measure. Doing this (see for example Figures 4.1, 4.3, and 4.4), show clear cases of misalignment before application of the new technique and none afterwards.

The other two ways of evaluating the results are numerical. Table 4.1 presents statistics on the successes and failures of the algorithm. These are evaluated for pairs that overlap but are not aligned by the pairwise algorithm. A success is declared if constraints are generated and verified by the new technique. A failure is declared otherwise. Combining the raw numbers, the algorithm has a 60% success rate on this preliminary test set. One of the eight failures has an image pair overlap of 17.8% of the image area, but examining these images shows no common constraints in the

Set	images	pairs	successes	failures	Max overlap
1	5	3	1	2	0.4%
2	8	5	3	2	2.8%
3	8	8	6	2	17.8%
4	9	4	2	2	1.1%

Table 4.1: Summary of the success rate for 4 image data sets. The column labeled "images" indicates the number of image used in the test. The column labeled "pairs" indicates the number of pairs that overlap but do not have a pairwise registration result; these are candidates for the addition of constraints. The column labeled "successes" indicates the number of pairs for which a constraint set was generated and verified. The column labeled "failures" indicates the number of failed pairs. Finally, the last column indicates the fraction of image overlap of the failed pairs.

set	The pair		Others	
	before	after	before	after
1	2.43	0.88	0.69	0.71
2	1.92	1.30	0.83	0.83
3	1.16	1.10	0.97	0.97
4	0.82	0.48	0.58	0.58

Table 4.2: Effect of the constraints addition on the average error in alignment (CEM). The second and third columns show the average CEM before and after application of the new technique for unregistered, but overlapping image pairs. The fourth and fifth columns show the average CEM before and after for registered image pairs.

area of overlap. All of the other seven failures have image overlaps less than 3% of the image area. These clearly cause no significant misalignment in the mosaics.

The final numerical evaluation is to study the centerline error measure (CEM). Table 4.2 and Table 4.3 show two important results. First, as expected, the CEM for previously unaligned image pairs improves substantially. Second, the CEM for previously aligned image pairs does not increase significantly. In other words, the addition of the new constraints corrects the bad results without biasing the previously good results.

Overall, these experiments have shown the efficacy of our new techniques both visually and numerically.

set	The pair		Others	
	before	after	before	after
1	5.48	0.93	0.79	0.84
2	3.40	1.56	0.99	0.99
3	1.97	1.80	1.31	1.32
4	1.26	0.58	0.59	0.59

Table 4.3: Effect of the constraints addition on the maximum error in alignment (CEM). The second and third columns show the maximum CEM before and after application of the new technique for unregistered, but overlapping image pairs. The fourth and fifth columns show the maximum CEM before and after for registered image pairs.

4.5 Summary and Conclusions

We have presented a technique that constructs mosaics of sparsely-overlapping image sets, and demonstrated this technique on retinal fundus image sets. The primary technical contribution of this chapter is the use of uncertainty to drive the formation of new matching constraints between unregistered image pairs in the context of multi-image alignment. This is particularly important for low overlap images and images with sparse feature sets such as the blood vessels of retinal images. We have shown that this technique corrects misalignments in sparsely overlapping image sets without sacrificing overall accuracy.

From the applications perspective this technique is crucial for moving retinal image registration and mosaicing techniques out of the laboratory and into the clinic. Mosaics for most image data sets can be constructed seamlessly without using the new techniques; a few can not. These few, however, can ruin the utility of a system. By adding the diagnostic and refinement capabilities of the covariance-driven techniques described here, we have developed a reliable, autonomous retinal image registration and mosaicing system.

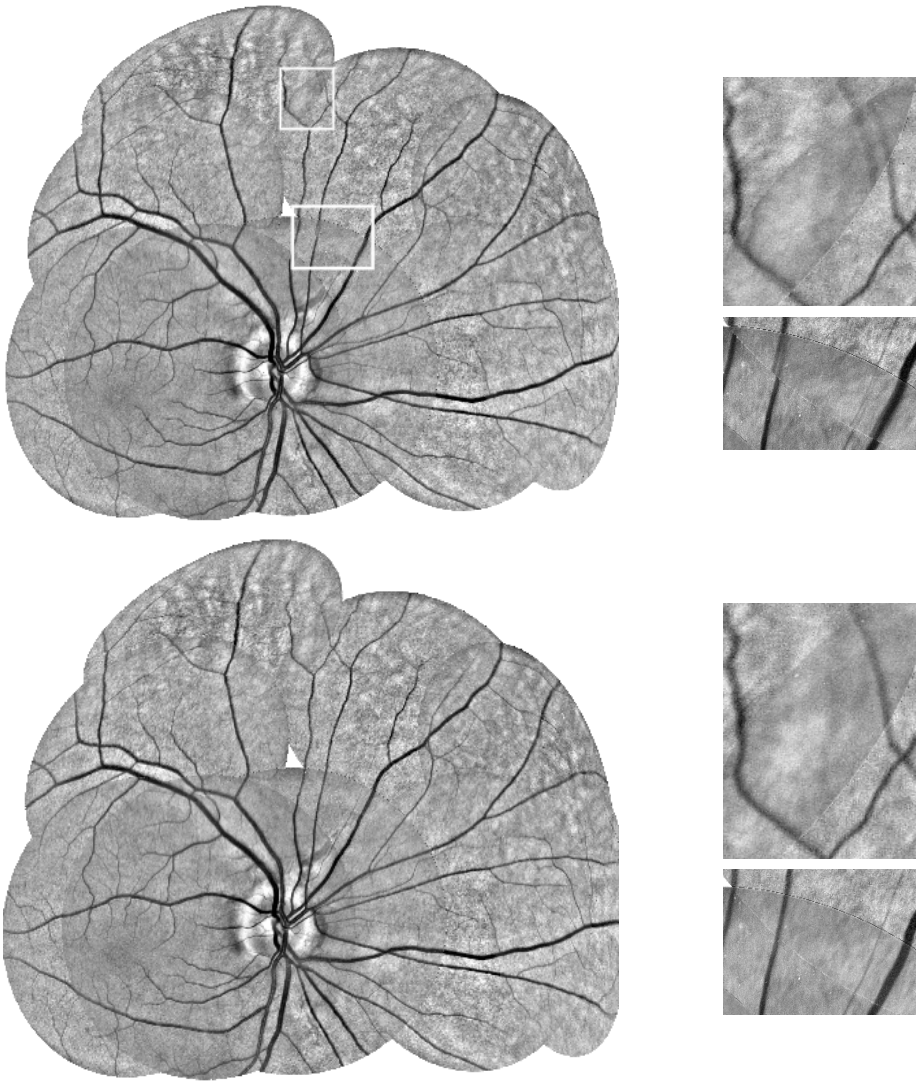


Figure 4.3: A subset of a 30-field image set illustrating misalignment before application of the new covariance-driven constraint-generation technique (top) and correction afterwards (bottom). The image sections shown to the right show regions of misalignment (top) and their correction (bottom) using the new technique.

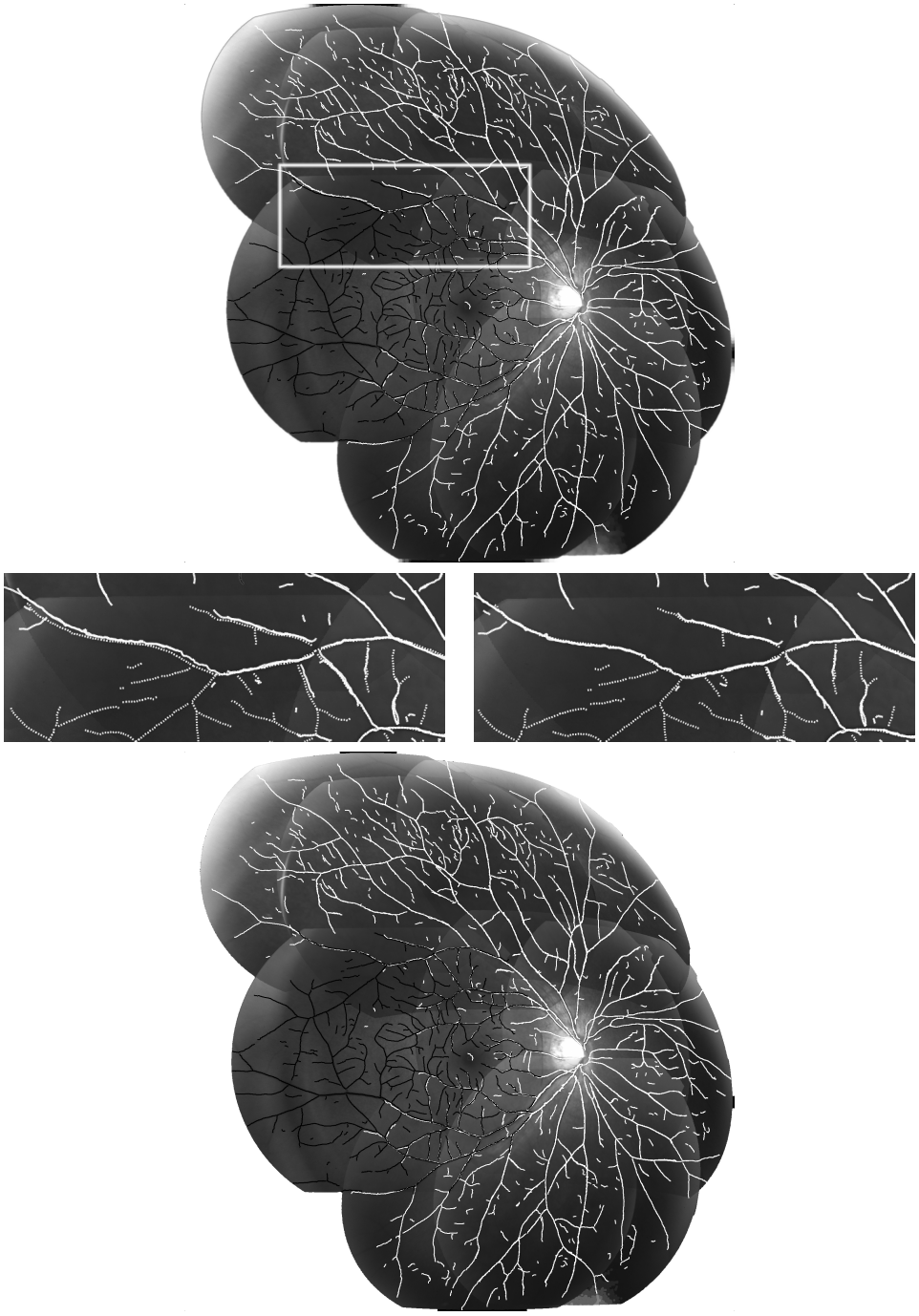


Figure 4.4: A third example result. The top shows a mosaic with centerline points drawn on it. The solid lines are centerline points from one image whereas the dotted ones are from anther. The highlighted rectangular region is shown in expanded view (center left) and the corrected version is shown next to it (center right). The final mosaic is shown at the bottom.

CHAPTER 5
Generalized Dual-Bootstrap ICP

5.1 Overview

Building on the success of the RIVERS system for retinal image registration, we would like to take a step further — to align images taken of natual scenes and to tackle the retinal images failed by the RIVERS without having the need to extract features specific to retinal images. More importantly, we would like to extend the success of our work to different imaging modalities, such as Infrared, and MRI.

The issues lie in both the photometric and geometric aspects of the images to be registered. Photometrically, images are acquired by different modalities, with different illumination, or with occlusions. Geometrically, images are related by some unknown translation, rotation, and scaling, as well as camera and projective distortions, which require non-linear transformation models, such as homography with radial lens distortion coefficients. In addition, some images may have low overlap, making it more difficult to determine the transformation.

In this chapter, we focus on these issues and tackle the three technical challenges described in Chapter 1 — initialization, estimation, and decision. In particular, the estimation of transformation is built with the extension of the Dual-Bootstrap algorithm (demonstrated in Figure 5.1). New feature extraction and decision criteria are developed to address these issues.

5.1.1 Related Techniques

The general literature on image registration was presented in Chapter 2. We focus our attention here on four classes of methods that appear most appropriate for the general-purpose registration problem being addressed.

- **ICP and feature-based methods** There are two common problems with feature-based methods and ICP: (1) ICP has a narrow domain of convergence and therefore must be initialized relatively accurately; and (2) feature extraction can be unreliable and overly sensitive to the choice of parameters and the

image content. (We will address both issues.) As an improvement, Expectation Maximization (EM) algorithms have broader domain of convergence. While this approach can improve the robustness and reliability of ICP, it has not been shown to work on data sets as difficult as our test suite. We discuss this further in Section 5.7.

- **Keypoint Indexing Methods** While this approach has all the steps needed for a general-purpose registration system, it has important limitations for the types of image pairs considered in this chapter. Most importantly, experiments on our test suite show that keypoint indexing and matching methods only produce a small number of correct matches, occasionally none and sometimes fewer than 10 out of the top 100 matches. Finding an accurate estimate in this case is either impossible, or requires an expensive, combinatorial search of possible match sets. It is possible that more sophisticated keypoint detection, description and matching methods will make this overall approach viable for such challenging image pairs, but this is not the direction taken in this chapter.

- **Mutual Information** For the current problem mutual information is less useful than it might first appear. First, algorithms based on MI [59, 164] are only estimation techniques and include neither initialization nor decision techniques. The former is often not a problem in medical applications, and the latter is often left to human judgment. Moreover, MI minimization procedures are often quite sensitive to local minima, a crucial concern when aligning low-overlap image pairs.

- **Direct Methods** These techniques require good initialization, although coarse searches of the translation-only parameter space have sometimes been proven effective [179]. Moreover, direct methods have not yet been demonstrated on registration problems involving substantial scale and orientation differences between images. Finally, no decision criterion is associated with these methods.

5.1.2 Approach

We propose an end-to-end registration algorithm — actually a system of algorithms — for aligning pairs of images using parametric global transformation models, building heavily on existing techniques. We use our test suite both to carefully evaluate the main components of our algorithm and to demonstrate its overall effectiveness. The primary novelties of our work are in the construction of the overall algorithm, the design and interrelationship of the components, and the validation of the system and the components. As an illustration of this, the alignment process is driven by image features extracted using auto-correlation matrices [182], which have been widely used in Harris corner detectors and its variants [12, 80, 143]. Our implementation produces edge-like as well as corner-like features distributed throughout the images, even in extremely low-contrast regions. By stressing dense coverage of features in images we ensure that features will nearly always be available, trusting the robustness of the matching and estimation algorithms, as well as the decision criteria, to avoid the effects of inconsistencies between images. Thus, the novelty is in tailoring feature extraction for general-purpose registration rather than feature extraction per se.

As mentioned earlier, the three primary components of the system are the initialization algorithm, the estimation technique, and the decision criteria:

- The initialization method uses extraction and matching of keypoints [122]. But unlike current techniques it does not attempt to combine matches in any way. Instead, each match is used to generate an initial similarity transformation that is accurate only in a small region surrounding the matched keypoints from each image. There are two reasons for this. First, as suggested above, for challenging image pairs a relatively small fraction of keypoint matches is correct — too small for effective use of minimum-subset random sampling search techniques. Second, for the keypoint matches that are correct, our growth-and-refinement based estimation procedure usually aligns the images accurately. Both reasons will be illustrated experimentally.

- The estimation technique starts with the initial local regions and associated transformation estimates, treating each one individually and in succession.

The goal for each region and estimate is to expand the region to cover the entire overlap between images while refining the estimate each time the region changes. This can be thought of as keeping the estimate close to the optimum as the problem grows in complexity. At the same time, as more information is incorporated, it may be possible to switch to a higher-order transformation model that more accurately describes the mapping of larger image regions. These intuitions are realized by generalizing the Dual-Bootstrap ICP algorithm, originally proposed for retinal image registration [199]. Several generalizations are needed to make this work. The most important is the use of generic features, as discussed above, in place of detected blood vessels to drive registration. Other enhanced techniques include bidirectional, across-scale matching between images, region growth in both images, use of a more general model selection technique, and an estimation technique that accounts for variations in feature location uncertainty. It is interesting to note that these generalizations allow the new algorithm to handle some retinal image pairs that the original algorithm could not.

- The decision technique determines if an estimate generated by the Dual-Bootstrap algorithm is indeed a correct alignment of the two images. The technique combines novel decision criteria that measure accuracy, consistency and stability in the alignment.

The overall procedure takes an ordered sequence of initial estimates (generated from keypoint matching) and applies Dual-Bootstrap ICP to each in succession. Following the convergence of each Dual-Bootstrap estimate the decision criteria are applied. If the estimate passes these tests, it is accepted as correct and the two images are considered registered. Otherwise, the process continues to the next initial keypoint estimate. It stops with failure if none succeed. The complete algorithm will be referred to as the Generalized Dual-Bootstrap ICP — GDB-ICP for short.

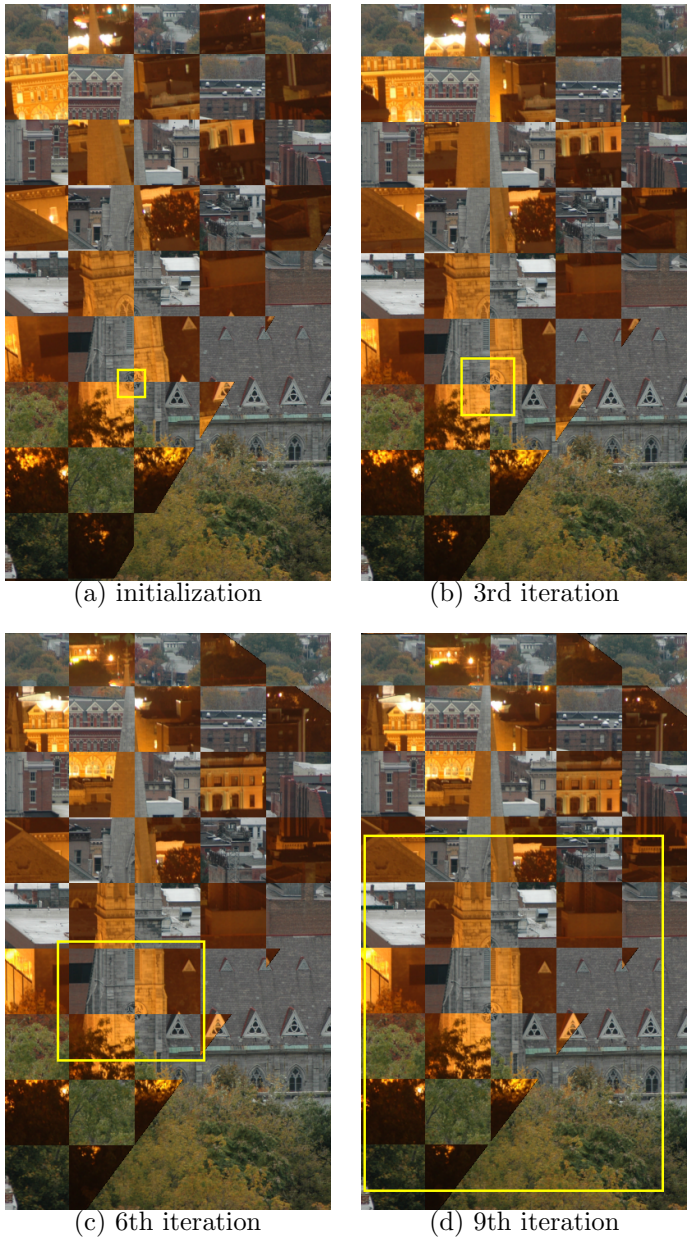
The remainder of this chapter describes the details of GDB-ICP and then evaluates it on our test suite of challenging image pairs (Figure 1.9, Figure 1.10, Figure 1.11, and Figure 1.12). The experimental analysis demonstrates the overall performance of the algorithm and explores the significance of many of the individ-

ual components and design decisions. Importantly, this includes testing on not only the image pairs that overlap and should therefore be registered but also all possible image pairs, including those with no overlap. The chapter concludes with a discussion of the strengths and limitations of GDB-ICP, and its relationship to other techniques.

5.2 Initialization

Our initialization method is based on Lowe's multiscale keypoint detector and SIFT descriptor [122]. This has proven to be the most effective in the experimental evaluation of [142]. We have used both our own implementation and the one used in the above evaluation with approximately equal success. We have also used affine-invariant multiscale Harris corners [143], and again the differences in results were minor. The results reported here used the implementation from [142].

Here is a summary of the initialization method. Each Lowe keypoint is a local maximum of the magnitude of the Laplacian-of-Gaussian operator in both spatial and scale dimensions. A neighborhood is established at each keypoint, with size determined by the keypoint scale and orientation determined by the local gradient direction. The intensity gradient vectors within this neighborhood are collected in histograms to form a 128-component SIFT descriptor vector. All SIFT vectors are stored in a spatial data structure, one for each image. Keypoint matching between images occurs by taking the keypoints from one image and using their descriptors to index into the data structure for the other image. The descriptor distance, measured as the Euclidean distance between vectors, is computed for each candidate match. The two closest matches for each descriptor are found and the ratio of the distances to the closest and second closest is calculated. Lowe establishes a upper bound threshold on this ratio of 0.8, and considers only matches below this threshold. Instead, we sort the matches by the ratio and test the top N in order, stopping when GDB-ICP produces an estimate that the decision criteria accept or when all N matches have been tested unsuccessfully. We have found $N = 50$ to be a good, conservative choice. The initial similarity transformation is established from the positions of the two keypoints, the orientations of their dominant intensity



(a) initialization (b) 3rd iteration

(c) 6th iteration (d) 9th iteration

Figure 5.1: Example steps of the Dual-Bootstrap growth and refinement process on the Day-Night Summer pair. The checkerboard images above show the alignment results and bootstrap region for iterations 0, 3, 6 and 9. The yellow rectangle outlines the bootstrap region in one image. Within this region, which is where the computation is focused, the alignment is generally accurate. Outside the region, especially for the small regions early in the computation, the alignment tends to be inaccurate. As the bootstrap region expands, more and more of the images are accurately aligned. The final alignment is shown in Figure 5.9.

Figure 5.2: Initial keypoint match and side-by-side alignment for one of our winter-summer pairs. The image region on the right has been scaled by a factor of 2.25, and there are substantial illumination, blurring, and changes (snow) between the regions.

gradients, and their scales (Figure 5.2). The initial bootstrap region is a square centered at each keypoint location, with half width $30 + 3s_i$, where s_i is the scale of the ith keypoint in the image where it is detected. This width setting ensures that there are enough constraints for Dual-Bootstrap to start successfully for keypoints of various scales. One initial region is generated on each of the two images, centered at the keypoint location. Together with the initial transformation, they are provided as input to the Dual-Bootstrap.

One important consideration is why should keypoint descriptors designed only for invariance to linear changes in intensity be useful in multimodal image registration. Indeed, as we will see experimentally, keypoint matching is the least successful component of our algorithm. On the other hand, our algorithm relies on finding only one correct keypoint match, and in generating this match, the descriptors do not have to match exactly. Instead they must only be distinct from other matches. Furthermore, *local* intensity differences between analogous keypoints in different modalities are sometimes well-approximated by linear transformations. These two observations explain why matching of Lowe keypoints successfully generates at least a few correct matches on a variety of multimodal pairs.

5.3 Feature Extraction

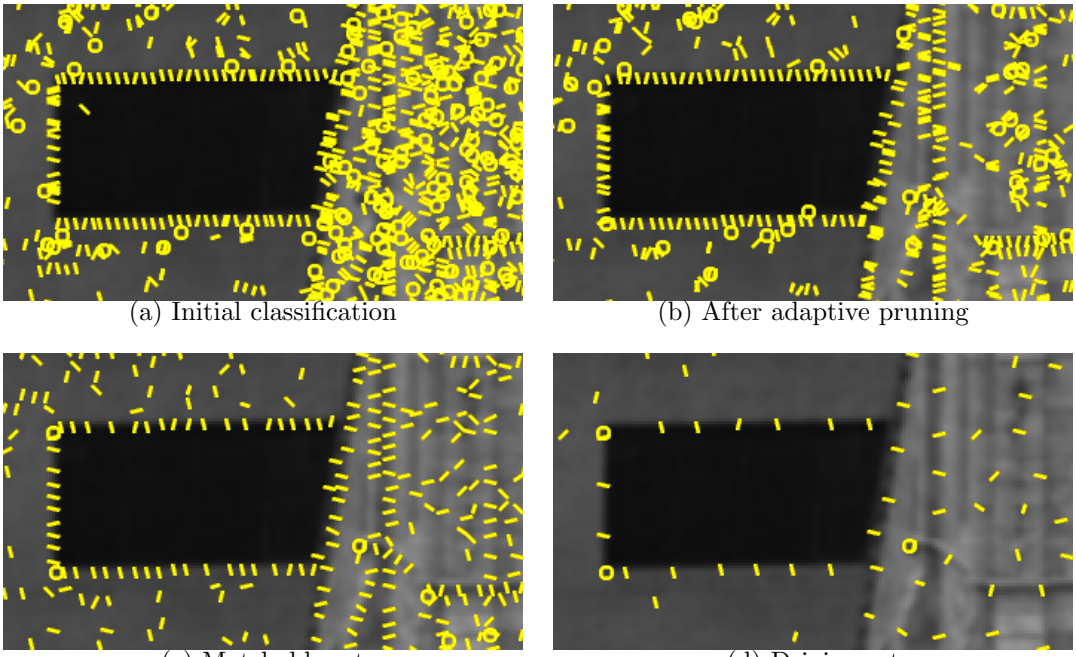
The Dual-Bootstrap procedure is feature-driven for two reasons. First, matching image features provides direct measurement of the geometric alignment error.

Figure 5.3: Examples of substantial variations between image regions due to illumination differences, scale differences, and physical changes (snow).

This is needed to compute the statistics that drive the growth, model selection and decision processes. The second reason is motivated by the changes seen between images that must be aligned. Much of the image texture and details may change between images — e.g. snow covering rooftops, leaves missing, textured regions falling into deep shadows — but structural outlines usually remain unchanged (Figure 5.3). These outlines, large or small, can be captured by properly-extracted features.

As discussed in the introduction, the primary consideration in designing the feature extraction method is not obtaining a complete or a perceptually-significant set of features. It is solely to extract features useful to drive the alignment process. Because of this, our goal is a feature-extraction method that (a) does not depend on thresholds or parameters that must be tuned to individual image content, (b) produces repeatable features and (c) distributes features throughout the image rather than concentrating them in the image regions of highest contrast.

Two different types of features are extracted — corner points and face points.



(a) Initial classification (b) After adaptive pruning

(c) Matchable set (d) Driving set

Figure 5.4: Intermediate results of the feature extraction process: (a) initial corners (shown with circles) and face points (shown with line segments along the eigenvector corresponding to the larger eigenvalue) after applying the threshold t_m (only one of every three is shown in this display because they are so dense), (b) corners and face points after adaptive local pruning, (c) "matchable" corners and face points, and (d) "driving" corners and face points.

Corner points provide two constraints on registration, whereas face points, because of tangential position ambiguity, provide only one. On the other hand, face points are more densely distributed. We will evaluate the contribution of each feature type to the registration process in the experimental section. Both corners and face points are extracted in scale-space with scale increasing in half-octave steps (e.g. scales $1, \sqrt{2}, 2, 2\sqrt{2}, \ldots$), with no combination across scales. Features at all scales are used simultaneously during registration. The following details of the feature-extraction algorithm are presented for a single scale.

We use a single response measure for both feature types. At each pixel location $\mathbf{x}$ at scale σ, the intensity gradient, $\nabla I(\mathbf{x})$, is computed. A weighted neighborhood

outer product (auto-correlation) matrix is then computed,

$$\mathbf{M}(\mathbf{x}) = \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} w(\mathbf{x}-\mathbf{y})\ (\nabla I(\mathbf{y}))(\nabla I(\mathbf{y}))^{\top}, \tag{5.1}$$

where w is a Gaussian weight function with standard deviation σ and the neighborhood size is approximately 3σ. The value of $\mathbf{M}(\mathbf{x})$ is normalized across scale σ using the normalization factor as in [118]. Next, the eigen-decomposition is computed: $\mathbf{M}(\mathbf{x}) = \sum_{i=1,2} \lambda_i(\mathbf{x})\mathbf{\Gamma}_i(\mathbf{x})\mathbf{\Gamma}_i(\mathbf{x})^{\top}$, with $\lambda_1(\mathbf{x}) \leq \lambda_2(\mathbf{x})$. Potential *corners* are at pixels where $\lambda_1(\mathbf{x})/\lambda_2(\mathbf{x}) > t_a$. This criterion is similar to the Harris corner detector [80]. Potential *face points* are at pixels for which $\lambda_1(\mathbf{x})/\lambda_2(\mathbf{x}) \leq t_a$. Decision value t_a has been experimentally set to 0.1; although the choice of values is not crucial. A strength is assigned to each point: $m(\mathbf{x}) = \text{trace}(\mathbf{M}(\mathbf{x}))$.

The next step is designed to make the final selection of feature points adaptive to local image contrast. First, a very low threshold, $t_m = 1$, is applied to the strength to eliminate points that are obviously noise. The result is illustrated in Figure 5.4a. The next step, local pruning, starts by computing the median μ_m and median absolute deviation (MAD) [200] σ_m of the values of $m(\mathbf{x})$ in a coarse set of overlapping neighborhoods (30×30 pixels) throughout the image. Then, pixels with strength $m(\mathbf{x}) < \mu_m + 0.5\sigma_m$ are eliminated from further consideration. As the final preliminary step, non-maximum suppression is applied at each point — in 2D for potential corner points and in 1D, along direction of the eigenvector $\mathbf{\Gamma}_2$, for face points — and surviving point locations are interpolated to subpixel accuracy. Figure 5.4b shows an example result.

The final steps are to extract the actual features, generating the sets of corners and face points independently for the use in registration. Points surviving the previous step are sorted by $m(\mathbf{x})$ values. The highest strength point is labelled as a feature. Remaining points within a small neighborhood are eliminated. This continues until a maximum number of features — determined from the size of the images — is extracted or the list is exhausted. This produces a set of features which we call the *matchable features* (Figure 5.4c). The procedure is repeated starting from the noise thresholding step with the elimination neighborhood doubled in size,

half the allowed number of features, and a threshold requiring $m(x) > 2t_m$. This produces a set of *driving features* (Figure 5.4d). As will be explained later, driving features are transformed and matched against matchable features, similar to [189]. Since driving features must pass stricter criteria than matchable features, it is less likely that a driving feature in one image will be missed as a matchable feature in the other due to random effects.

An example of extracted driving and matchable feature sets at different scales is shown in Figure 5.5. Features are spread throughout the image and summarize the local image structure: a corner is placed in regions containing substantial intensity variations in all directions; a face point occurs where the region contains variation in one direction.

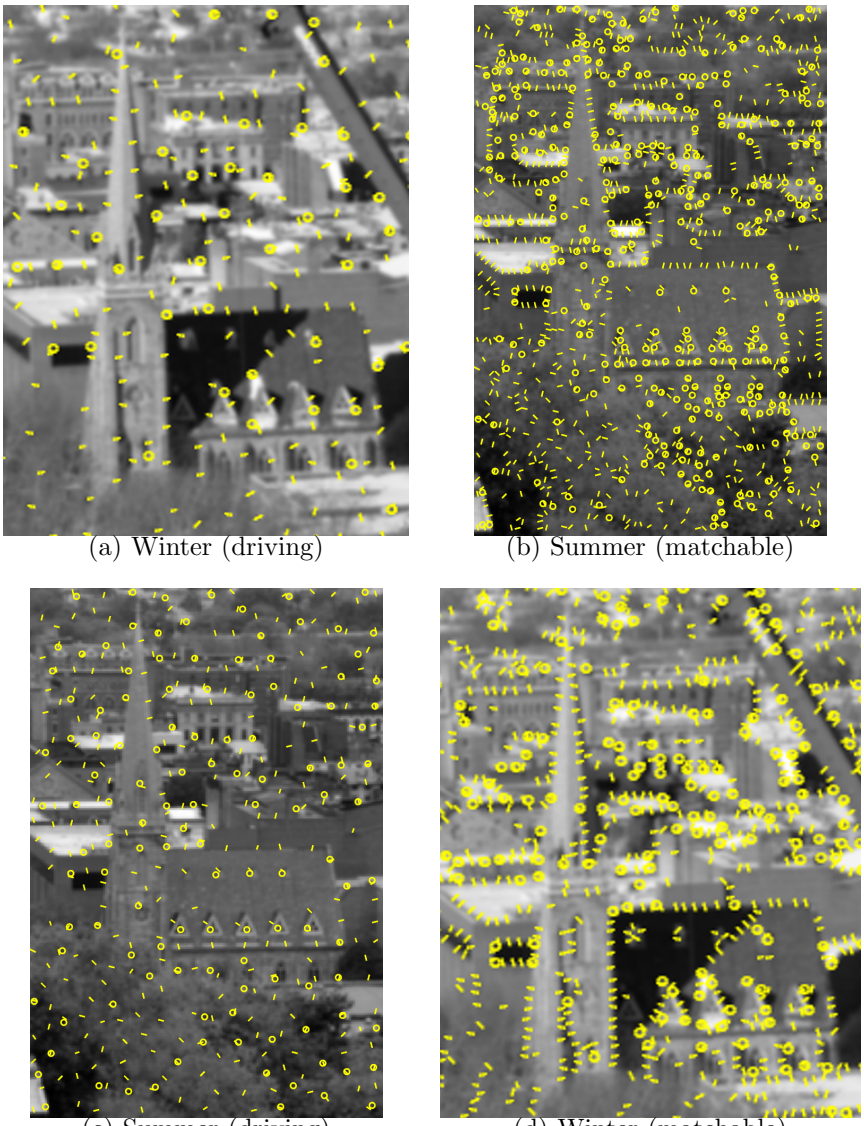
5.4 GDB-ICP Estimation

The estimation step of the Generalized Dual-Bootstrap ICP (GDB-ICP) algorithm starts with an initial similarity transformation generated from keypoint matching, together with the initial bootstrap region surrounding the keypoint location in each image. The algorithm iterates steps of (1) refining the current transformation inside the current bootstrap region R, (2) applying model selection to determine if a more sophisticated model may be used, and (3) expanding the region, growing inversely proportional to the uncertainty of the mapping on the region boundary (Figure 5.1). The entire algorithm is outlined in Figure 5.6.

5.4.1 Notation

The two images are I_p and I_q. The matchable corner and face points are $\mathcal{P}_c = \{\mathbf{p}_c\}$ and $\mathcal{P}_f = \{\mathbf{p}_f\}$ from I_p and $\mathcal{Q}_c = \{\mathbf{q}_c\}$ and $\mathcal{Q}_f = \{\mathbf{q}_f\}$ from I_q. Driving feature sets are subsets of $\mathcal{P}_c$, $\mathcal{P}_f$, $\mathcal{Q}_c$ and $\mathcal{Q}_f$. Points from all scales are combined to form these sets. Abusing notation, $\mathbf{p}$ and $\mathbf{q}$ represent both the feature and its location. Each feature has associated with it a feature scale, s, at which it was detected, and each face point has a normal direction, $\boldsymbol{\eta}$.

The *forward* transformation of point location $\mathbf{x}$ from I_p onto I_q is $\mathbf{T}(\mathbf{x}; \boldsymbol{\theta}^{pq})$, where $\boldsymbol{\theta}^{pq}$ is the parameter vector to be estimated. An estimate is $\hat{\boldsymbol{\theta}}^{pq}$, and its



(a) Winter (driving) (b) Summer (matchable)

(c) Summer (driving) (d) Winter (matchable)

Figure 5.5: Example of the "matchable" and "driving" features and bidirectional matching. In matching, "driving" features from the (cropped) winter image (a) are mapped onto the summer image and matched against the "matchable" features (b). Symmetrically, "driving" features from the summer image (c) are mapped onto the winter image and matched against the "matchable" features (d). Although not illustrated in this figure, matching may occur between features at different scale-space scales.

1. Extract keypoints and features from images I_p and I_q (Sec. 5.2 and 5.3).
2. Match keypoints from I_p to I_q and rank-order the matches (Sec. 5.2).
3. **Do**
   (a) Choose next keypoint match and generate the initial transformations and bootstrap regions (Sec. 5.2).
   (b) **Repeat**
      i. Use the current model and parameter estimates $\hat{\boldsymbol{\theta}}^{pq}$ and $\hat{\boldsymbol{\theta}}^{qp}$ to generate the forward and backward match sets $\mathcal{C}_c^{pq}$ or $\mathcal{C}_f^{pq}$, $\mathcal{C}_c^{qp}$ and $\mathcal{C}_f^{qp}$ (Sec. 5.4.2).
      ii. Use the match sets to re-estimate the forward and backward transformation parameter estimates, $\hat{\boldsymbol{\theta}}^{pq}$ and $\hat{\boldsymbol{\theta}}^{qp}$, and their covariance matrices, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}^{pq}}$ and $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}^{qp}}$, for the current model and remaining higher order models (Sec. 5.4.3).
      iii. Use the estimates and covariance matrices to select the model for the next iteration (Sec. 5.4.4).
      iv. Grow the bootstrap regions in each image separately using the selected model, parameter estimates, and covariance matrices (Sec. 5.4.5).

      Until region growth, model selection and parameter estimation have converged.
   (c) Apply decision criteria using the final model, parameter estimates, covariance matrices, and face correspondence sets. If all pass accept parameter estimates $\hat{\boldsymbol{\theta}}^{pq}$ and $\hat{\boldsymbol{\theta}}^{qp}$ and terminate (Sec. 5.5).
4. **While** fewer than N keypoint matches have been tried.
5. Terminate with failure

Figure 5.6: Outline of the Generalized Dual-Bootstrap Algorithm

covariance matrix is $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}^{pq}}$. The *backward* transformation from I_q onto I_p is $\mathbf{T}(\mathbf{x}; \boldsymbol{\theta}^{qp})$, with an estimate $\hat{\boldsymbol{\theta}}^{qp}$ and covariance estimate $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}^{qp}}$. Finally, the regions over which the transformation are being estimated are called the "bootstrap" regions, and are denoted by R_p on image I_p and R_q on image I_q. Each is defined as an axis-aligned rectangular box on its own image.

5.4.2 Matching Within the Bootstrap Region

The transformation is refined within current bootstrap regions R_p and R_q, ignoring everything else in the two images. Recall that in standard ICP the current

transformation is used to generate a new set of correspondences, in turn these correspondences are used to generate a new transformation, and this process iterates. By contrast GDB-ICP proceeds to model selection and region growing before selecting a new set of matches.

GDB-ICP uses bi-directional matching. This provides more constraints and helps to produce more numerically-stable estimates, especially for small bootstrap regions. A driving feature $\mathbf{p}$ is mapped from I_p to I_q, producing $\mathbf{p}' = \mathbf{T}(\mathbf{p}; \hat{\boldsymbol{\theta}}^{pq})$. The three closest matchable features (of the same type) to $\mathbf{p}'$ are found in I_q, and the best matching feature, $\mathbf{q}$, is chosen from among these three based on a similarity measure described below. The correspondence pair $(\mathbf{p}, \mathbf{q})$ is added to match sets $\mathcal{C}_c^{pq}$ (corners) or $\mathcal{C}_f^{pq}$ (faces). Reversing order, the pair $(\mathbf{q}, \mathbf{p})$ is also added to either $\mathcal{C}_c^{qp}$ or $\mathcal{C}_f^{qp}$. The same procedure is used in the opposite direction to generate correspondences between driving features from I_q and matchable features from I_p. Once these matches are generated, the forward and backward transformation estimates $\hat{\boldsymbol{\theta}}^{pq}$ and $\hat{\boldsymbol{\theta}}^{qp}$ are calculated. Since these use the same set of constraints, just with the feature roles reversed, the two transformation estimates are close to being inverses of each other (typical root mean square error of 0.1 pixels or less). The remainder of the discussion focuses on the calculation of $\hat{\boldsymbol{\theta}}^{pq}$ using $\mathcal{C}_c^{pq}$ and $\mathcal{C}_f^{pq}$.

A similarity measure is used both in choosing between matches for a feature and in weighting the chosen match during estimation. For corners it depends only on the feature scales, but for face points it depends on orientations as well. For a feature point $\mathbf{p}$, let $s_{\mathbf{p}'}$ be the feature scale after the transformation is applied. For a face point, let $\boldsymbol{\eta}_{\mathbf{p}'}$ be the transformed normal direction. The similarity measures for a prospective match with $\mathbf{q}$, with scale $s_{\mathbf{q}}$ and (for a face point) normal $\boldsymbol{\eta}_{\mathbf{q}}$, are

$$w_c = \min(s_{\mathbf{p}'}/s_{\mathbf{q}}, s_{\mathbf{q}}/s_{\mathbf{p}'}) \qquad \text{and} \qquad w_f = \min(s_{\mathbf{p}'}/s_{\mathbf{q}}, s_{\mathbf{q}}/s_{\mathbf{p}'}) \cdot |\boldsymbol{\eta}_{\mathbf{p}'}^\top \boldsymbol{\eta}_{\mathbf{q}}|$$

for corners and face points, respectively. This biases the selection toward features at similar (mapped) scales and orientations, and allows for contrast reversals in face point matches as well.

5.4.3 Estimation

Estimation is applied to the current model and, as the basis for model section (Sec. 5.4.4), to higher-order models under consideration. This section describes estimation for a single model.

Before defining the transformation estimate objective function, we need to define the error distances. For corner points these are Euclidean distances, whereas for face point these are normal distances:

$$d_c(\mathbf{p}, \mathbf{q}; \boldsymbol{\theta}^{pq}) = \|\mathbf{T}(\mathbf{p}; \boldsymbol{\theta}^{pq}) - \mathbf{q}\|/s_{\mathbf{q}} \qquad \text{and} \qquad d_f(\mathbf{p}, \mathbf{q}; \boldsymbol{\theta}^{pq}) = |(\mathbf{T}(\mathbf{p}; \boldsymbol{\theta}^{pq}) - \mathbf{q})^\top \boldsymbol{\eta}_{\mathbf{q}}|/s_{\mathbf{q}}.$$

In each case, the distance is normalized by the scale at which the feature is detected, reflecting the fact that feature location uncertainty increases with increasing scale. This makes distances of features at different scales approximately comparable.

Combining the foregoing, the objective function for estimating transformation parameters $\boldsymbol{\theta}^{pq}$ from a fixed set of correspondences is

$$E(\boldsymbol{\theta}^{pq}; \mathcal{C}_c^{pq}, \mathcal{C}_f^{pq}) = \sum_{(\mathbf{p}_i, \mathbf{q}_i) \in \mathcal{C}_c^{pq}} w_{c;i} \rho(d_c(\mathbf{p}_i, \mathbf{q}_i; \boldsymbol{\theta}^{pq})/\sigma_c) + \sum_{(\mathbf{p}_i, \mathbf{q}_i) \in \mathcal{C}_f^{pq}} w_{f;i} \rho(d_f(\mathbf{p}_i, \mathbf{q}_i; \boldsymbol{\theta}^{pq})/\sigma_f) \tag{5.2}$$

where $\rho(\cdot)$ is the Beaton-Tukey [88, 137, 200] robust loss function:

$$\rho(u) = \begin{cases} \frac{a^2}{6}[1 - (1 - (\frac{u}{a})^2)^3], & |u| \leq a \\ \frac{a^2}{6}, & |u| > a \end{cases} \tag{5.3}$$

Following standard usage, the constant a is set to 4, which means that normalized alignment error distances beyond 4σ have a fixed cost. The parameters σ_c and σ_f are the robust alignment error scales (standard deviations) for the normalized distances of corners and faces.

Objective function (5.2) is minimized using the Iteratively-Reweighted Least-Squares (IRLS) technique from the robust statistics literature [137, 200], which alternates computation of a distance-based weight $w_{d,i}$ for each correspondence, i, based on fixed transformation parameters with weighted least-squares re-estimation

of the parameters from

$$F(\boldsymbol{\theta}^{pq}; \mathcal{C}_c^{pq}, \mathcal{C}_f^{pq}) = \sum_{(\mathbf{p}_i, \mathbf{q}_i) \in \mathcal{C}_c^{pq}} w_{d,i} w_{c,i} d_c^2(\mathbf{p}_i, \mathbf{q}_i; \boldsymbol{\theta}^{pq}) + \sum_{(\mathbf{p}_i, \mathbf{q}_i) \in \mathcal{C}_f^{pq}} w_{d,i} w_{f,i} d_f^2(\mathbf{p}_i, \mathbf{q}_i; \boldsymbol{\theta}^{pq}). \tag{5.4}$$

The distance-based robust weight factor for corners is $w_{d,i} = w(d_c(\mathbf{p}_i, \mathbf{q}_i; \boldsymbol{\theta}^{pq})/\sigma_c)/\sigma_c^2$, where $w(\cdot)$, derived from the Beaton-Tukey robust loss function, is

$$w(u) = \begin{cases} 1 - (\frac{u}{a})^2, & |u| \leq a \\ 0, & |u| > a \end{cases}.$$

An analogous computation produces the weights for face points. Normalization factors $1/\sigma_c^2$ (for corners) and $1/\sigma_f^2$ (for face points) make corners and face points comparable.

The robust standard deviation, σ_c, for corner matches is recomputed once (per correspondence set and per Dual-Bootstrap iteration) from the weights and current transformation estimate as

$$(\sigma_c)^2 = \sum_{(\mathbf{p}_i, \mathbf{q}_i) \in \mathcal{C}_f^{pq}} w_{d,i} w_{c,i} d_c^2(\mathbf{p}_i, \mathbf{q}_i; \hat{\boldsymbol{\theta}}^{pq}) \; / \sum_{(\mathbf{p}_i, \mathbf{q}_i) \in \mathcal{C}_c^{pq}} w_{d,i} w_{c,i},$$

with a similar computation for face matches. At the start of the Dual-Bootstrap procedure for a given initial transformation, the MUSE algorithm [147] is used to estimate σ_c and σ_f from the first set of matches, since weights are unavailable.

Finally, the computation of the weighted least-squares estimates from (5.4) and the associated covariance matrix of the parameter estimates, which is needed for the region growth and decision criterion, use standard techniques. In particular, for the transformation models that are unconstrained and linear in their parameters, $\hat{\boldsymbol{\theta}}$ is obtained in closed-form using linear weighted least-squares. The covariance matrix, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}^{pq}}$ of the estimate is the inverse of the Hessian of (5.4). Usually this must be multiplied by a noise variance term, but this is already built in to the distance-based robust weights $w_{d,i}$. For constrained or non-linear models, such as the planar homography or planar homography plus radial-lens distortion terms,

Levenberg-Marquardt [132] is used, with the pseudo-inverse of the Hessian giving the covariance matrix, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}^{pq}}$ (see [82, Ch. 4]).

5.4.4 Model Hierarchy and Model Selection

The goal of model selection is to select the model from a set (a hierarchy) of transformation models that best describes the current bootstrap region. As the region expands, model selection is applied to choose between the model used for the previous bootstrap region, and the remaining, high-order models. Model selection must be done carefully. Switching to a higher-order model too early, especially when the region is small and there are insufficient constraints, may lead to overfittings and distortions in the estimate. Switching too late causes an increase in mapping errors and a resulting increase in mismatches. In either case, incorrect model selection may drive the estimate into a local minimum representing an incorrect alignment.

Two different model hierarchies are used in GDB-ICP. One, used for retinal images, is a hierarchy moving from similarity to a reduced quadratic to a quadratic model (see [199] for details). The second, used for natural images, is a hierarchy of similarity, affine, homography, and homography plus radial lens distortion (HRD). The HRD model is defined as

$$\mathbf{T}(\mathbf{p};\boldsymbol{\theta}) = \mathbf{D}(\mathbf{T}_H(\mathbf{D}(\mathbf{p};k_p);\mathbf{h});k_q)$$

where $\boldsymbol{\theta}^\top = (\mathbf{h}^\top, k_p, k_q)$, $\mathbf{T}_H(\mathbf{x};\mathbf{h})$ is the usual planar homography ($\mathbf{h}$ is a 9-component vector formed from the 3×3 homography matrix), and $\mathbf{D}(\mathbf{x};k) = (1 + k\|\mathbf{x} - \mathbf{x}_0\|^2)\mathbf{x}$ is the radial distortion function, given image center $\mathbf{x}_0$, assumed to be the center of the pixel array. This model is important for accurate alignment of digital photographs taken with off-the-shelf cameras.

Model selection techniques have been studied extensively in the literature[27, 26, 99, 217]. Earlier version of the Dual-Bootstrap method [199] used a Bayesian technique derived in [26] that depends on computing the determinant of the parameter-estimate covariance matrix, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}^{pq}}$. For homographies this is problematic because $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}^{pq}}$ is not full rank. Rather than developing an appropriate projection onto a full-rank covariance, we have replaced the Bayesian criteria with a modified version of

Akaike Information Criteria (AIC), derived from the Kullback-Leibler measure, and found it to be quite effective. Using our robust objective function (5.2) and taking the advantage of having transformation estimates and match sets available in both directions, a second-order Akaike Information Criteria may be written as

$$\begin{aligned}\mathcal{I} = &-2\left[\mid \mathcal{C}_c^{pq} \mid \log(\sigma_c^{pq}) + \mid \mathcal{C}_f^{pq} \mid \log(\sigma_f^{pq}) + E(\hat{\boldsymbol{\theta}}^{pq}; \mathcal{C}_c^{pq}, \mathcal{C}_f^{pq})\right] \\ &-2\left[\mid \mathcal{C}_c^{qp} \mid \log(\sigma_c^{qp}) + \mid \mathcal{C}_f^{qp} \mid \log(\sigma_f^{qp}) + E(\hat{\boldsymbol{\theta}}^{qp}; \mathcal{C}_c^{qp}, \mathcal{C}_f^{qp})\right] + 2nl/(n-l-1),\end{aligned} \tag{5.5}$$

where l is the degrees of freedom in current model, $n = 2 \mid \mathcal{C}_c^{pq} \mid + 2 \mid \mathcal{C}_c^{qp} \mid + \mid \mathcal{C}_f^{pq} \mid + \mid \mathcal{C}_f^{qp} \mid$ is the effective number of constraints (each corner match provides with two constraints, while each face point match provides one), and the term $2nl/(n-l-1)$ adjusts for small-sample bias [27, p. 51].

Expression (5.5) is evaluated for each candidate model using a fixed set of matches found using the transformation estimate of the best model from the previous Dual-Bootstrap iteration. The final objective function value of (5.2) (after IRLS converges) is used for each model to evaluate expression (5.5). The model that minimizes (5.5) is chosen as the current model and its estimated parameters become the current parameters. Model selection is turned off once the selection procedure reaches the highest-order model.

5.4.5 Region Growth

Region growth, illustrated in Figure 5.1, is based on the uncertainty in the transformation estimate, represented by the covariance matrix $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}^{pq}}$. Expansion of the axis-aligned rectangle representing the current bootstrap region is inversely proportional to the transfer error — the error in applying the estimated transformation to points on the boundary of the bootstrap region. The following is a summary of the details of this procedure taken from [199]. Subsequently, a simple modification is given to make the algorithm more effective in registering image pairs with large scale variations.

Let the center of the bootstrap region be $\mathbf{y}_0$, let a point location centered on one of the four sides of the region be $\mathbf{y}$, and let $\eta_y = (\mathbf{y} - \mathbf{y}_0)/\|\mathbf{y} - \mathbf{y}_0\|$ be

the outward-pointing normal to the rectangle. The mapping error covariance at the mapped point $\mathbf{y}' = \mathbf{T}(\mathbf{y}; \hat{\boldsymbol{\theta}}^{pq})$ is computed from the Jacobian of the mapping, $\mathbf{J}_{\mathbf{y}} = \partial \mathbf{T}(\mathbf{x}; \boldsymbol{\theta}) / \partial \boldsymbol{\theta}$ evaluated at $\mathbf{x} = \mathbf{y}$ and $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}^{pq}$, together with the covariance of the transformation parameters:

$$\boldsymbol{\Sigma}_{\mathbf{y}'} = \mathbf{J}_{\mathbf{y}} \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}^{pq}} \mathbf{J}_{\mathbf{y}}^{\top}. \tag{5.6}$$

The outward growth rate is inversely proportional to the error variance in the mapped outward normal direction, η'_y:

$$\delta_y = \beta \frac{(\mathbf{y} - \mathbf{y}_0)^{\top} \eta_y}{\max(1, \eta_y'^{\top} \boldsymbol{\Sigma}_{\mathbf{y}'} \eta'_y)},$$

where $\beta = 2.0$ is a fixed constant. The new center of the side is given by $\hat{\mathbf{y}} = \mathbf{y}_0 + (1 + \delta_y)(\mathbf{y} - \mathbf{y}_0)$. The new region is obtained after all the side centers are updated with the above method.

The extension made here is to form and grow bootstrap regions independently in each image, I_p and I_q. The initial regions are determined from the separate keypoint locations and scales, as described in Section 5.2. The above procedure is applied separately for the two regions at each Dual Bootstrap iteration separately. These regions are implicitly kept relatively consistent through the use of bi-directional matching, which keeps the transformations close to inverses of each other and the covariance matrices commensurate with each other. Keeping separate regions in the two images is important for handling large scale differences between images.

5.5 Decision Criteria

Once the GDB-ICP refinement procedure just described expands to cover the apparent overlap between images (based on the estimated transformation) and the refinement process has converged, the final alignment is tested for correctness. If this confirms that the transformation is correct, the images are considered to be aligned, and the algorithm stops. Otherwise, the next keypoint match is tested

using GDB-ICP.

Three tests — accuracy, stability and consistency — form the decision criteria. The tests are applied in each direction using the final match sets. A transformation that passes all three tests in both directions is accepted as correct.

Accuracy is measured as the weighted average error $\zeta_e(\hat{\boldsymbol{\theta}}, \mathcal{C}_f)$, computed on the final face matches, $\mathcal{C}_f$. Face points are used because their positions (along the normal direction) are more accurate than corner points. Using the measures introduced above, accuracy is

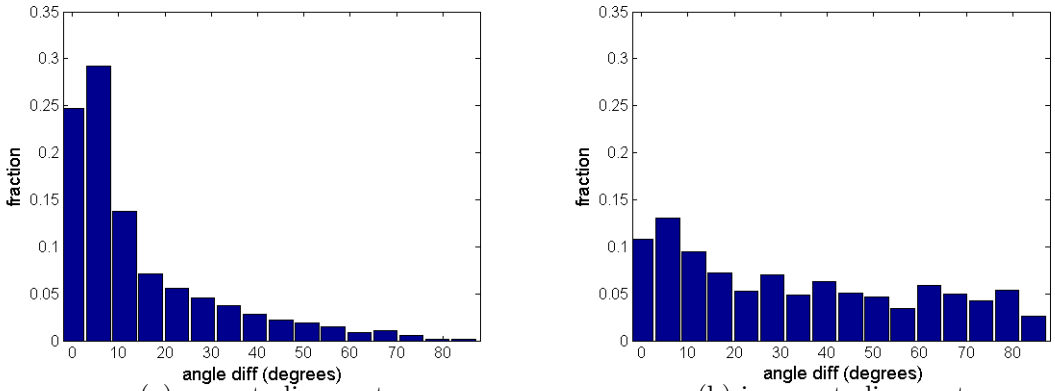
$$\zeta_e(\hat{\boldsymbol{\theta}}^{pq}, \mathcal{C}_f) = \Big(\sum_{(\mathbf{p}_i, \mathbf{q}_i) \in \mathcal{C}_f} w_{f,i} w_{d,i} |d_f(\mathbf{p}_i, \mathbf{q}_i; \hat{\boldsymbol{\theta}}^{pq}))| \Big) \Big/ \Big(\sum_{(\mathbf{p}_i, \mathbf{q}_i) \in \mathcal{C}_f} w_{f,i} w_{d,i} \Big). \tag{5.7}$$

Stability is measured by the error covariance — the mapping transfer error introduced in the context of region growth in Section 5.4.5. To check this, points are uniformly sampled in the overlap area between aligned images. For each sample point $\mathbf{y}_i$, the mapping error covariance $\boldsymbol{\Sigma}_{\mathbf{y}'_i}$ is computed from (5.6). The overall measure is $\zeta_t(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}) = \max_i \operatorname{trace}(\boldsymbol{\Sigma}_{\mathbf{y}'_i})$. This is particular effective at avoiding incorrect low-overlap transformations.

The consistency measure is derived from the orientation differences of the face point match set $\mathcal{C}_f$ after the application of the transformation estimate $\hat{\boldsymbol{\theta}}$. These differences, measured in absolute angle difference, are put into a histogram $h(\hat{\boldsymbol{\theta}}, \mathcal{C}_f)$ of the range $[0, \pi/2]$. (Absolute angle differences greater than $\pi/2$ are subtracted from π, effectively accommodating intensity reversals.) If the transformation is incorrect, this angle difference will tend toward being uniformly distributed, whereas if the images are well-aligned, the histogram will tend to have a strong peak near 0 degrees (Figure 5.7). The consistency measure is based on the Bhattacharyya measure against an exponential distribution ($\lambda = 4.7$ dictates 70% of the face point matches have orientation differences no greater than 10 degrees). This exponential distribution, denoted as e, is represented as a second histogram. Then the consistency measure $\zeta_c(\hat{\boldsymbol{\theta}}, \mathcal{C}_f)$ is

$$\zeta_c(\hat{\boldsymbol{\theta}}, \mathcal{C}_f) = 1 - \sum_i \sqrt{h_i(\hat{\boldsymbol{\theta}}^{pq}, \mathcal{C}_f) e_i} \tag{5.8}$$

To make a decision with these measures — ζ_e, ζ_t and ζ_c — lower and upper



(a) correct alignment (b) incorrect alignment

Figure 5.7: Example histograms of the orientation difference of a set of face point matches from correct (a) and incorrect (b) alignments using the Summer-Winter pair.

thresholds are introduced for each: $Z_e^L \leq Z_e^H$ for ζ_e, $Z_t^L \leq Z_t^H$ for ζ_t, and $Z_c^L \leq Z_c^H$ for ζ_c. When $\zeta_e \leq Z_e^L$, $\zeta_t \leq Z_t^L$ **and** $\zeta_c \leq Z_c^L$, the transformation estimate is accepted as correct. When $\zeta_e > Z_e^H$, $\zeta_t > Z_t^H$ **or** $\zeta_c > Z_c^H$, the transformation is rejected. Otherwise, the transformation is saved. If all initial transformations have been tested and none have been accepted, the saved transformation with the minimum value of alignment error ζ_e is accepted. If there are no saved transformations, the algorithm rejects the image pair, indicating that it can not be aligned. Transformations tend to fall into the "saved" category for image pairs that involve significant changes or that can not be precisely aligned using the final transformation model.

These thresholds are fixed at $Z_e^L = 1$, $Z_e^H = 2$, $Z_t^L = 0.3$, $Z_t^H = 1$, $Z_c^L = 0.09$, and $Z_c^H = 0.2$ for all experiments here. For efficiency, the algorithm also applies a set of higher thresholds, starting after the third Dual Bootstrap iteration when the estimate has begun to stabilize, to identify and terminate estimates that are clearly wrong. We refer to this step as the *early termination criteria*.

5.6 Experiments

This section presents experiments designed to illustrate the overall performance of the GDB-ICP algorithm (Section 5.6.2), compare it to minimum-subset

random sampling methods (Section 5.6.3), and analyze in detail the most important aspects of the algorithm. The focus of the latter is on the success of growth and refinement (Section 5.6.4), the choices of features and matching criteria (Section 5.6.5), and the effectiveness of the final decision criteria (Section 5.6.6) — the newest aspects of the algorithm. See [199] for analysis that shows the significance of the Dual-Bootstrap refinement, growth and model selection procedures in the context of retinal image registration.

5.6.1 Data Set

All experiments are on the test suite of 22 image pairs discussed in the introduction. This suite was constructed from our own digital photographs, from pairs found on the web, and from challenging pairs suggested by colleagues. Many easier pairs have been left out in order to keep the tests manageable. As an example of this, we included one pair, with 2% overlap (the "Dashpoint" pair here), from the test suite of [25] (GDB-ICP registers all overlapping pairs from this suite). On the other hand, some types of pairs, such as PET-CT images, which have no common geometric structure, have been purposely left out. We discuss this more in Section 5.7. The results are clearly conditioned on the test suite, but the range of challenging pairs shown should be suggestive of the broad effectiveness of our algorithm. In order to allow the community to test GDB-ICP beyond the experiments presented here, an executable version of the software has been posted on the web.

The images range in size from 676×280 to 2500×2500. Image pairs overlap as little as 2%, differ in scale by a factor as high as 6.4, and differ in orientation by as much as 90 degrees. Five pairs are multi-modal (retina angiogram vs. red-free photograph, two infrared vs. video airport scenes, pan-chromatic and infrared satellite images, and proton density vs. T1 weighted brain MRI slices). Four pairs involve substantial illumination changes and two other pairs are of different seasons. The selection of scenes includes aerial, urban, landscape, indoor, and medical. On the Melanoma and EO-IR 1 pairs, the intensity of one image is negated before keypoint generation since SIFT is not invariant to intensity reversal. Finally, the retinal images involve quadratic transformations, whereas the others involve the use of the

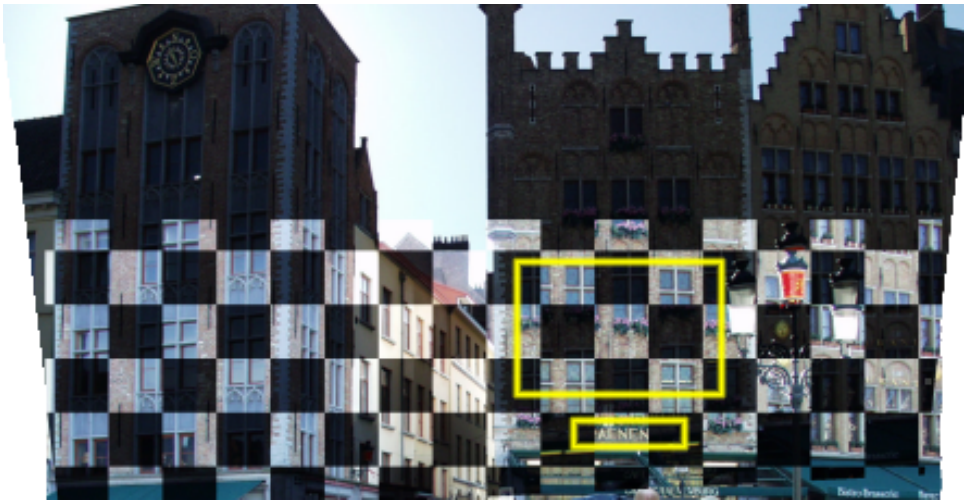
homography or the homography plus radial lens distortion (HRD) models. The choice of final model is specified by a command-line argument. All other parameter settings are fixed for these experiments.

5.6.2 Overall Results

GDB-ICP successfully aligned 19 of all 22 image pairs in our test suite with error less than a pixel. Success is defined here as no visible misalignments between homologous structures following application of the transformation, as judged independently by a graduate student who is not one of the co-authors of [236]. The successful transformations, one for each pair, are labeled as "verified" transformations to be used in subsequent experiments. Example alignments are shown in Figure 5.9 to Figure 5.13; complete results are posted at our web site, including animations. Interestingly, for pairs "Brugge", "Brugge Square" and "Brussels", the 10 degree-of-freedom "Homography plus Radial lens Distortion" (HRD) model eliminated small, but visible misalignments produced by using only a homography (see Figure 5.8 for details).

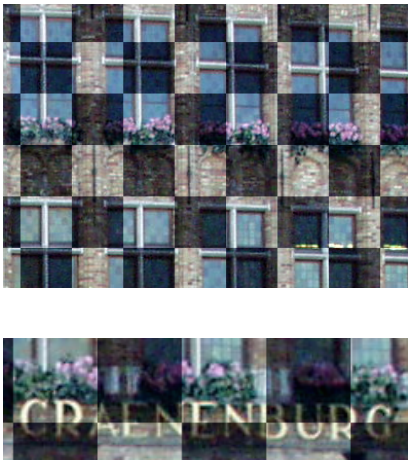
Table 5.2 shows, for each pair, the index number of the first keypoint match in the rank ordering for which the algorithm succeeded, the index of the same successful keypoint match among only those consistent with the verified transformation for which the algorithm succeeded, the final alignment error, and the chosen transformation model. A consistent keypoint match is somewhat arbitrarily defined to have a location error of less than 6 pixels, a scale ratio within the interval 0.8 to 1.25 (one step in scale space), and an orientation difference of 15 degrees, all computed following application of the verified transformation. Intuitively, these are matches that appear to be geometrically close to correct. The remaining keypoint matches are labeled "inconsistent". As can be seen from the table, in most cases a consistent match appears among the first five in the rank ordering and in 14 cases GDB-ICP successfully refined this initial transformation to a verified final transformation.

GDB-ICP failed for three pairs. In each case, manual specification of three initial correspondences in a small initial region and computation of an initial estimate of an affine transformation, following by application of the Dual-Bootstrap



(a) Checker mosaic

(b)Homography only



(c) HRD model

Figure 5.8: (a) shows a checker mosaic of the Brugge Square pair using the homography model, in which there are places, notably inside the boxes, that have misalignments. Column (b) shows zoom-in of the boxes. Column (c) shows the same area after using the Homography with Radial lens Distortion (HRD) model. The window frames and letters are now well aligned. The resulting alignment is is subpixel accurate.

growth and refinement procedure led to a verified transformation. This indicates that the failures are caused by keypoint detection and matching, by keypoint-based initialization, or by the early stages in the Dual-Bootstrap growth and refinement procedures. In one case in particular — Capital Region — the projective distortions are too severe be handled starting from a local similarity transformation.

Image sizes and timing results are summarized in Table 5.1. Clearly algorithm speed is mostly affected by image size and matching difficulty. The failures and the image pairs requiring testing of all 50 keypoint matches (because no match provided results below the lower decision thresholds) are the only ones other than the huge "Satellite" pair requiring more than a minute.

5.6.3 Comparison to Keypoint Matching Algorithms

As one indication of the significance of these results, the publicly-available code for the Autostitch keypoint matching algorithm of [23] (with default parameters) produced five alignments ("Boston", "Boston Library", "Eiffel", "Brugge Square", and "Brussels"). The latter three have visible misalignments, partly due to the fact that the homography is insufficient for these pairs [11]. On the other 17 pairs Autostitch failed altogether. We obtained similar results with our own implementation using RANSAC and other random-sampling-based algorithms ([147, 218]). The failures are due to both the small number and the small fraction of consistent keypoint matches, as shown in the last three columns of Table 5.2.

5.6.4 Success of the Growth and Refinement Procedure

The following experiment shows the effectiveness of starting from individual keypoint matches rather than combining them, as in a random-sampling approach. We use the top 50 keypoint matches of the 19 pairs that GDB-ICP aligned. The GDB-ICP estimation process is applied to each of these keypoint matches, without any decision criteria. The resulting "test" transformation estimate is then compared to the verified transformation. Those that agree to within an average distance of less than 2 pixels are considered correct.

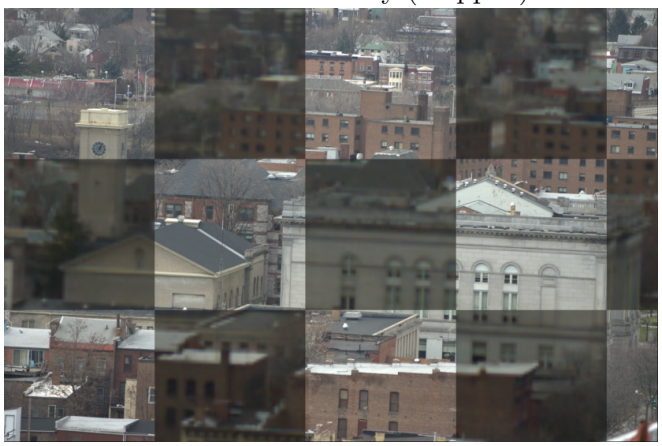
[11] Though more complex model can be used with keypoint-matching algorithms, the increase in the degrees of freedom adversely affects the effective use of RANSAC.

Image Pair Name	Image Dimension	On the successful keypoint match		Grand Total	
		# of Iterations	Time	# of Keypoint matches tried	Time
Boston	1712×1368	14	12	1	12
Boston Library	1504×1000	13	5.7	1	5.7
Brain T1 to PD	512×512	12	2.0	3	2.9
Brugge Square	1712×1368	17	52	2	58
Brugge Tower	1712×1368	20	52	1	52
Brussels	1712×1368	17	37	2	41
Capital Region	1712×1368	NA	NA	50	1225
Day-Night	1000×1504	24	38	2	39
Dashpoint	2048×1536	12	10	1	10
Eiffel	1712×1368	16	15	2	19
EO- IR 1	300×236	15	6.8	50	127
EO- IR 2	676×280	NA	NA	50	51
Extreme Zoom	1504×1000	25	34	50	761
Grand Canyon 1	1184×780	15	9.6	2	12
Grand Canyon 2	900×568	16	26	50	592
Melanoma	1156×880	14	13	50	58
Retina	1600×1200	17	9.9	5	24
Satellite	2878×2878	18	40	4	122
Whiteboard	1504×1000	19	9.5	4	12
White Tower	1504×1000	18	21.5	2	23
Winter-Summer	1504×1000	19	30	1	30
Winter Day-Summer Night	1504×1000	NA	NA	50	300

Table 5.1: Timing results in seconds. The first two columns are the image-pair name and the dimension of the larger image. The next two columns are the number of iterations and the time that the Dual Bootstrap growth-and-refinement procedure spent on the keypoint match that led to the successful alignment. The last two columns are the total number of keypoint matches tried and the total time used before GDB-ICP terminates. The performance is measured on a Pentium 4 3.2GHz PC with 2GB memory.

White Tower

Winter-Summer Day (cropped)



Extreme Zoom (ratio 6.4, cropped)

Figure 5.9: Final alignment Checkerboard images

Brugge Square (cropped)

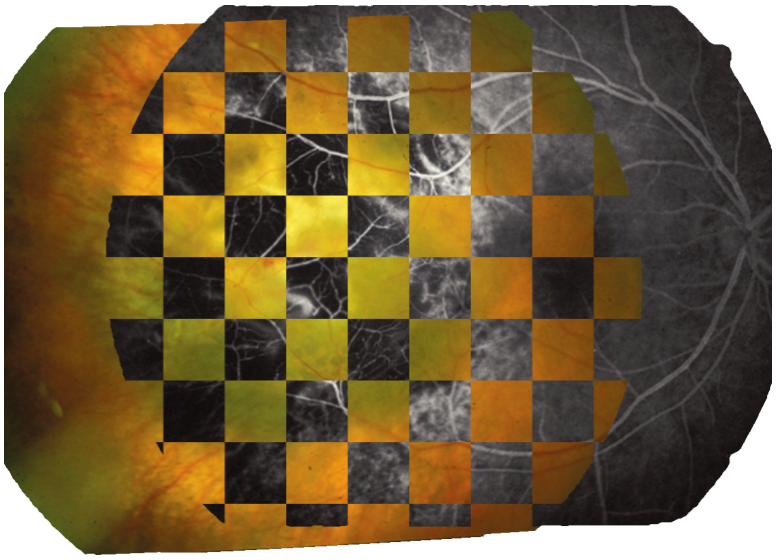
Day-Night Summer (cropped)

Figure 5.10: Final alignment Checkerboard images

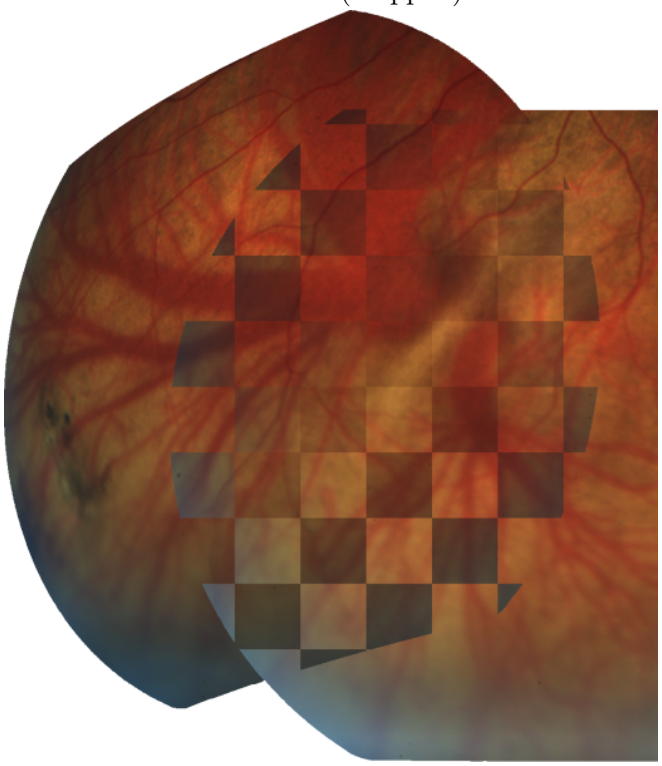
Dashpoint(overlap 2%, cropped)

Grand Canyon 2

Figure 5.11: Final alignment Checkerboard images

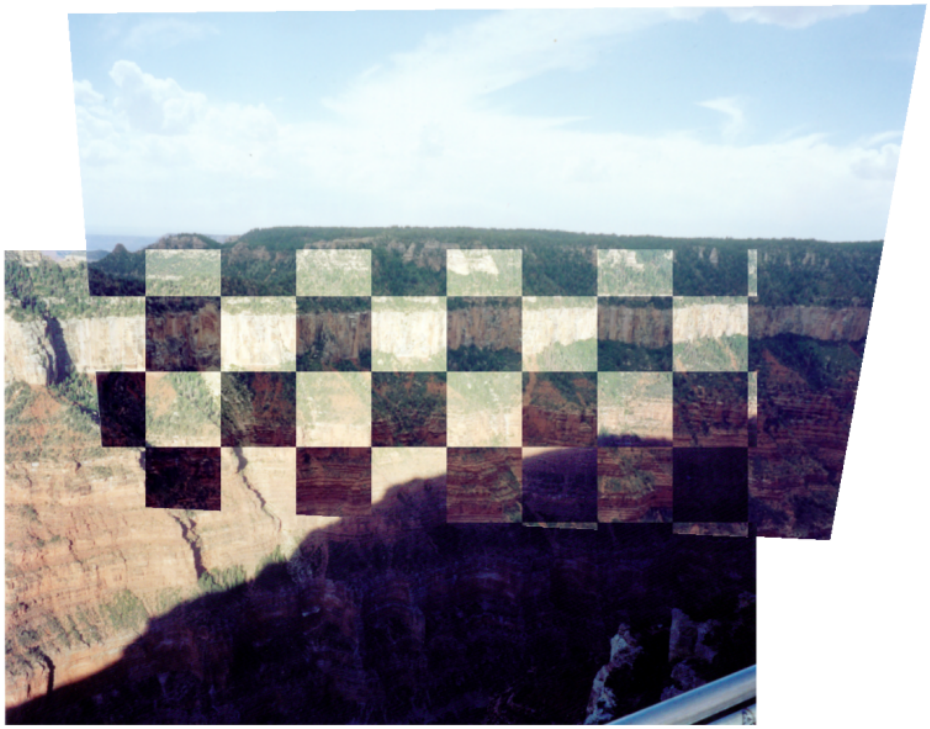


Melanoma (cropped)

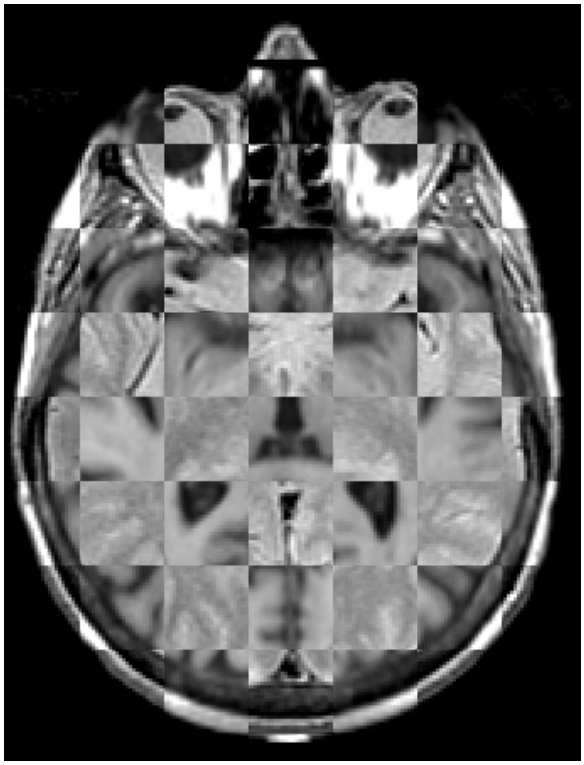


Retina (cropped)

Figure 5.12: Final alignment Checkerboard images



Grand Canyon 1



Brain PD-T1

Figure 5.13: Final alignment Checkerboard images

Image pair name	With GDB-ICP				Should RANSAC be used		
	1st succ. in all matches	1st succ. in only cons. matches	Final align-ment error	Final model	Total # of matches	pos. ≤ 6 pixels	ratio
Boston	1	1	0.36	HRD	1228	506	41%
Boston Library	1	1	0.32	HRD	1030	252	40%
Brain T1 to PD	3	2	0.98	H	40	2	5%
Brugge Square	2	1	0.43	HRD	758	287	37%
Brugge Tower	1	1	0.49	HRD	390	178	45%
Brussels	2	1	0.35	HRD	2728	1213	44%
Capital Region	NA	NA	NA	HRD	176	22	12%
Day-Night (Summer)	2	1	0.62	HRD	124	16	12%
Dashpoint	1	1	0.30	HRD	160	27	16%
Eiffel	2	1	0.30	HRD	442	117	26%
EO - IR 1	2	1	1.59	HRD	62	3	4.8%
EO - IR 2	NA	NA	NA	H	46	0	0%
Extreme zoom	1	1	0.33	HRD	448	89	19%
Grand Canyon 1	2	1	0.88	HRD	90	40	44%
Grand Canyon 2	5	1	1.20	HRD	54	14	26%
Melanoma	27	5	1.88	Q	14	2	14%
Retina	5	0[1]	0.58	Q	102	14	13%
Satellite	4	2	0.26	HRD	2152	989	46%
White board	4	2	0.20	HRD	342	96	28%
White Tower	2	1	1.14	HRD	58	7	12%
Winter-Summer (Day)	1	1	0.60	HRD	204	15	7%
Winter Day-Summer Night	NA	NA	NA	HRD	NA	NA	NA

Table 5.2: Summary statistics on all pairs from our data set. The 2nd to 5th columns show the performance of GDB-ICP: The index (starting from 1) of the first successful keypoint match in the rank-ordered list of all keypoint matches, the index of the same successful keypoint match in only the consistent keypoint matches , the final alignment error, and the choice of final transformation model — Homography (H), Homography with Radial lens Distortion (HRD), and Quadratic (Q). The last three columns show about keypoint statistics should minimum-subset random sampling techniques be used: the total number of keypoint matches (Lowe's similarity ratio < 0.8), the number whose positions are within 6.0 pixels following application of the verified transformation, and the ratio between the two. The last column — inlier ratio — indicates how many trials must be performed before obtaining a minimum set with only inliers. [1]On the "Retina" pair, GDB-ICP succeeded on an "inconsistent" keypoint match — one with 8.0 pixels of position error.

Among the 19 pairs of images, there are 950 keypoint matches in total, 363 are "consistent" and 587 "inconsistent". Among the 363 consistent ones, 282 led to correct final transformations, while among the 587 labeled "inconsistent", 28 of these led to correct final transformations, resulting in a total of 310 correct alignments. Examination of the 28 shows that the estimation procedure recovered from initial location errors as high as 12 pixels and orientation differences as much as 18 degrees.

To interpret the significance of these results, based on a probability of $P_a = 282/363 \approx 0.78$ of succeeding from a consistent keypoint match, the overall probability of GDB-ICP producing the correct alignment given n good matches is $1 - (1 - P_a)^n$, which is 0.99 when there are just $n = 3$ consistent matches. By contrast, minimal subset random sampling techniques require 4 matches just to instantiate a transformation. Clearly GDB-ICP can succeed despite an extremely small number of keypoint matches.

5.6.5 Choice of Features and Matching

The next set of experiments evaluates several variations on the choice of features, the scale of the features, and the directionality of matching. This is important to show the influence of these design decisions on the performance of the overall algorithm. These experiments show that face points and bidirectional matching are indispensable.

Just as in the previous test, we evaluate all 50 keypoint matches from the 19 pairs GDB-ICP succeeds upon. We study the change in the aforementioned 310 successes with changes in the feature extraction and matching. We also determine whether one of these changes causes the entire GDB-ICP to fail on a pair on which it originally succeeded.

The results are summarized in Table 5.3, which shows several important results. First, using corner points alone without faces results in a 44% drop in the number of successful initial keypoint matches, and a loss of 9 successful pairs (4 pairs if the "correct" tolerance is raised from 2 to 5 pixels). Apparently, corners are not widely-enough and densely-enough distributed for GDB-ICP to succeed consistently on our challenging test suite. Using faces alone, the drop is only 3%, and no pairs

are lost.

Interestingly, using forward matching alone instead of bidirectional matching causes the loss of 28% of the successful keypoint initializations and the three of the most difficult pairs (EO-IR 1, Extreme zoom, and Melanoma). In a related result, not shown in Table 5.3, the percentage of driving features that are mapped to within 2 standard deviations of their corresponding matchable features, thereby creating "inlier" correspondences, ranges from 58% to 83%. This indirectly justifies (a) the ability of GDB-ICP to adapt to substantial differences between images and (b) the decision to push feature extraction toward covering as much of an image as possible, trusting the rest of the algorithm to automatically determine which features are consistent between images.

The final test, shown in the last four rows of Table 5.3, explores multiscale feature extraction. Using scale 1.0 (standard deviation of Gaussian smoothing) results in a loss of nine initializations, but one image-pair — the Melanoma pair. When the single scale at which the features are extracted is increased, the success rate drops quickly. Finally, when using features combined across scales, similar to the scale-space detection technique of many keypoint matching algorithms [143, 117], there is a 13% drop in the number of successful initializations, but no loss of any pairs.

variations	succ. init.	succ. pairs
corners and faces	310	19
corners only	174	10
faces only	300	19
driving only	314	18
matchable only	300	19
forward matching	224	16
scale 1.0	291	18
scale 1.4	175	8
scale 2.0	96	5
combined across scales	272	19

Table 5.3: GDB-ICP success numbers based on varying the feature set and the matching. The 2nd column is the number of initializations (keypoint matches) that led to successful alignments, while the 3rd column is the pairs for which at least one initialization succeeded.

5.6.6 Decision Criteria

To analyze the strength of the three-component decision criteria, we compared them with simplified versions. The results show that all three components of the decision criteria are necessary, that bi-directional decision increases robustness, and that the complete decision criteria are effective in distinguishing correct alignments from incorrect ones, even in the presence of low overlap, scale differences and physical changes.

One of the simplified versions is the use of alignment error alone, a natural measure for registration based on geometric constraints and the one used in [32, 199] for retinal image registration. We then considered the importance of the three criteria by leaving each out in turn. Finally, we considered the effect of several other aspects of the decision criteria. In these experiments, all $42 \cdot 41 = 1722$ possible ordered pairs of images are used, with both orderings used for each pair because each can produce different initial keypoint matches (see Section 5.2 for details), and therefore different initializations. The final model is always the homography (for speed considerations). Alignments passed by the modified decision criteria were examined by a graduate student (not one of the co-authors of [236]) to determine correctness. This turned out to be crucial because this test discovered some small overlaps in our image set that we did not realize existed. Based on this human judgment and based on our verified results, the decisions made by GDB-ICP under various decision criteria could be classified as True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN). No changes were made in the parameter values of the decision criteria throughout the experiment.

The results are summarized in Table 5.4. The first observation concerns the effectiveness of the full decision criteria. The six false negatives come from the three pairs which GDB-ICP known to fail on. The three false positives are image pairs that appeared locally consistent, with one of the regions having very low contrast (one example shown in Figure 5.14). On the other hand, most such pairs are rejected. In fact, 99.8% of the incorrect pairs are rejected. Stated even more strongly, among the 1671 rejected pairs there are $1671 \times 50 = 83,550$ incorrect initializations, including many with low overlap, all of which are rejected.

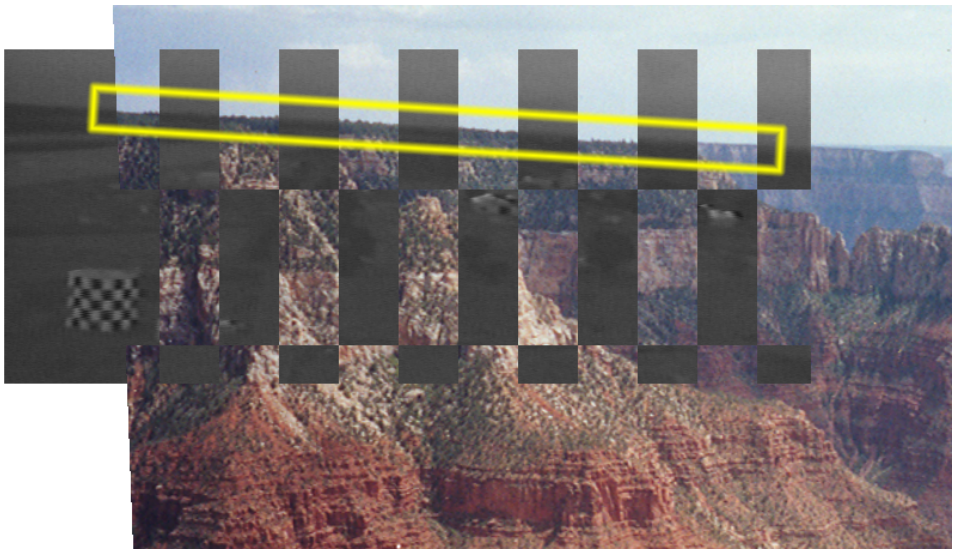


Figure 5.14: A cropped example of the false positives when applying the decision criteria to all possible pairs. The yellow box outlines the area where the alignment locks on. Note that the grayscale image to the left is a frame of a video sequence with very low contrast.

	TP	TN	FP	FN
Full	42	1671	3	6
Accuracy alone	41	1515	158	8
No Stability	42	1614	60	6
No Orientation	42	1646	28	6
No Accuracy	42	1638	36	6
Only forward	42	1641	33	6

Table 5.4: Effects of varying the decision criteria when applying GDB-ICP to all possible pairs. TP stands for True Positives, TN for True Negatives, FP for False Positives, and FN for False Negatives. See text in Section 5.6.6 for details.

As seen in the next four rows of the table, when using only part of the criteria, the number of false positives increases significantly — jumping to 158 for use of the accuracy measure only, but with fewer when two of the three measures are used. These false positives are due to locally-consistent structures, especially near image boundaries (producing apparent low-overlap between images). These experiments show convincingly that all three decision criteria are important. Finally, when the decision criteria are applied only in the forward direction — from the moving image to the fixed image — the false positive rate increases substantially.

5.7 Discussion

The experiments on our challenging test suite have demonstrated that GDB-ICP is a powerful registration algorithm, capable of aligning a wide variety of image pairs. Overall, our experience with this data set and with other pairs shows that GDB-ICP succeeds when keypoint matching produces a small number of consistent matches, when there is sufficient common structure between the images to drive the dual-bootstrap process and the decision criteria, and when the similarity transformation is a reasonable initial model. In this section, we examine this success, using the experiments to show how the design of the algorithm allows it to handle the image registration challenges outlined in the introduction. We also discuss some limitations of the algorithm. We conclude this section by re-examining several individual components of the algorithm.

Our experiments have shown that GDB-ICP can succeed with as few as one consistent keypoint match and with as little as 58% correct matches between the driving features of one image and the matchable features of the other. Remembering that matchable features must pass less-stringent tests than driving features and recalling that contrast reversals are ignored by the matching process, this result explains why the algorithm does so well with substantial changes in image illumination and structure, and even changes in modality. The tolerance for differences in feature extraction and matching is allowed in the algorithm because the decision critera can be trusted to reject nearly all incorrect alignments. The effectiveness of the decision criteria is also crucial to the algorithm's success in handling low-overlap pairs. Using the criteria, GDB-ICP accepts a small number of correct, low-overlap alignments while rejecting the extremely large number of low-overlap alignments generated by incorrect keypoint matches. Finally, the ability to generate matches across scales is crucial to handling substantial differences in scale.[12] While another recent algorithm has shown the ability to handle large scale variations [53], it has not be demonstrated in as challenging a context as our test suite.

Despite the demonstrated success, GDB-ICP does have limitations:

[12] After these tests were completed we tested a pair with a zoom factor of 9.5 and GDB-ICP succeeded.

- It can not handle extreme appearance differences between image pairs. In our test suite this is mostly due to initial keypoint matching, but we anticipate that the algorithm will fail on other multimodal pairs such as PET-CT pairs, where it is unlikely that the features will capture enough structural similarity. Still, GDB-ICP did succeed on all but one of the multimodal pairs in our test suite, because multimodal images often do have sufficient common structure. Intuitively, the structural and textural differences between the color image and the fluorescein angiogram in the "Melanoma" pair put it near the limit of what GDB-ICP can handle.

- Currently, the decision criteria do not eliminate incorrect alignments of an image-wide repetitive structure such as a checkerboard. On the other hand if only a moderate fraction of a scene involves repetitive structure, the decision criteria make the right decision: although incorrect alignments in the repetitive region will appear accurate, these produce inconsistent matches image-wide. An example of this occurs in the "Brussels" pair of our test suite.

- As a 2d registration algorithm, GDB-ICP currently only tolerates a small amount of parallax. In a similar vein, it is currently limited to global transformation models.

- There is no convergence proof in the overall algorithm, just as there is no convergence proof in the ICP using anything but Euclidean match distances. In practice, however, on thousands of tests, GDB-ICP has always converged. One reason for this that both region growth and model selection are monotonic.

- Finally, GDB-ICP, while consistently running in less than a minute for two mega-pixel image pairs, is still somewhat slow.

All of these issues are topics of our ongoing work.

Finally, we make a few observations about the individual components of GDB-ICP:

- The experiments show the importance of using multiscale face point (edge-like) features during the alignment process instead of more sparse features such as

corners, even when corners are detected at multiple scales.

- While it is straight-forward to replace Lowe's LoG keypoint detector and SIFT descriptor vector with other current keypoint techniques, it would be surprising if this would improve keypoint matching substantially on our data set. Still, a thorough exploration of this question is worthy of future study.

- Our earlier work in the context of retinal image registration showed the advantages of the Dual-Bootstrap ICP growth and refinement procedure over robust ICP when starting from a single initial correspondence [199]. Intuitively, the combination of re-estimation in the bootstrap region and model selection keeps the estimate close to the local minimum while gradually increasing the problem complexity through region growth. In contrast, EM-ICP and RPM [42, 73] attempt to disambiguate matches and estimate the transformation through simultaneous, image-wide consideration of multiple matches per feature. These have not yet been proven to handle low image overlap and a large percentage of features that have no match whatsoever. Our informal experience has been that achieving robustness to unmatchable features pushes the design of the cost function toward heavy reliance on the distance to the nearest match for each feature, significantly reducing the importance of considering multiple matches per feature. A definitive answer to this question is beyond the scope of this chapter.

- The axis-aligned, rectangular shape of the region is a simple, efficient representation. The new use of two independent regions introduced here is important for handling large differences in scale. Region models more sophisticated than our rectangular model could be developed — e.g. ones encompassing only the points where the trace of the transfer error covariance matrix is below a threshold — but the rectangular shape has proven sufficient for all our experiments and has not been the cause of algorithm failure. Note that other region growth techniques have recently been proposed in the literature [58, 197]. Ours differs in that its growth is monotonic and is controlled by the uncertainty in the transformation estimate.

- The three-part decision criteria have proven to be essential for handling the challenging image pairs studied here. Other techniques include Brown and Lowe's combinatorial analysis of keypoint matches [23] and Belongie's use of distance, brightness variation and bending energy for recognition [13]. Clearly, keypoint-based measures alone are insufficient. Measures based on intensity variation or gradient magnitude differences [186] are not appropriate for the range of appearance variation seen here. Finally, although our three-part criteria have proven highly successful, they are not perfect and further improvements are possible.

5.8 Summary and Conclusion

This chapter has presented the fully automatic Generalized Dual-Bootstrap ICP (GDB-ICP) image registration algorithm designed to handle a wide variety of image pairs, including those showing scale changes, orientation differences, low overlap, illumination differences, physical changes and different modalities. Building extensively on existing work, the algorithm is in fact a series of algorithms designed to work together to solve the problem. Extensive experiments on a 22 image-pair test suite representative of these challenges have shown the effectiveness of the design and demonstrated that a broadly-applicable, fully-automatic image registration is possible. The experiments have also highlighted areas of potential improvement. The most important of these is initialization, especially when there are large appearance variations between images, caused by physical or illumination changes or differences in image modalities. Despite this, the experiments reported here and experience by both our group and others who have tested the GDB-ICP executable demonstrate that it is effective enough for widespread use.

CHAPTER 6
Camera Location Estimation

6.1 Overview

In a new 3d-to-2d image registration problem, the location of a hand-held camera must be determined with respect to a 3d model augmented with image texture information. This requires registering the 2d image with the 3d model and estimating the camera projection matrix, which after decomposition, gives the location of the camera. With no assumptions on the cameras being used, the intrinsic camera parameters, as well as the lens distortion parameters, must be estimated in order to obtain an accurate estimate of the camera location. This is essentially calibrating the camera using the image.

In the previous work, 3d models are often constructed with pure range scans, i.e., without any texture information. When aligning a 2d image with such a 3d model, structural features are often used to drive the registration process. For instance, line segments are extracted and matched with the line segments detected in the image [120, 173, 194]. In recent advancements on the design of LiDAR range scanners, a pre-calibrated camera is embedded within the range scanner, sharing the same optical pathway [112]. Besides range data, the scanner also acquires a set of intensity images. Since the camera is pre-calibrated both intrinsically and extrinsically, it becomes a trivial task to construct a 3d model augmented with texture information. This additional texture information can be used to aid the 3d-to-2d image registration, especially in man-made environments, where smooth surfaces in the scene often have texture that can be used to distinguish one surface from another. In a similar track, Umeda *et al.* use laser reflectance images to aid image registration [227]. Here, since the scanner's images arre freely available and have similar general appearance[13] with the other camera's image, we employ these intensity images acquired from the pre-calibrated camera to drive the 3d-to-2d image

[13]We must stress that significant photometric differences, caused by physical changes, illumination changes, or occlusions, may exist and that our algorithm is designed to handle these differences.

registration.

6.1.1 Related Literature

The general literature on image registration is presented in Chapter 2. We focus our attention here on three classes of methods for 3d-to-2d image registration.

- **Structural Features** Several methods are based on extraction and matching of line segments [120, 173, 194]. They rely on the detection of parallel lines in the image and presume the radial lens distortion is known a prior. Both of these restrict the applicability of these methods. Troccoli and Aleen proposed a method based on shadow matching [221], which requires knowledge of the sun position at the moment when the image is taken.

- **Keypoint** A second approach is based on model-to-image keypoint matching [158, 183]. (For details on extraction and matching of keypoints, please refer to Section 2.1.1.3.) This requires obtaining a sufficient number of correct keypoint matches to accurately estimate and verify the 3d-to-2d projection. Similar to 2d-to-2d registration, this can be problematic when there are substantial viewpoint, illumination and structural changes between the model and the test image.

- **Region Growing** A third category of approaches, which includes ours, is based on region-growing [60, 181]. In the closest work, Fraundorfer and Bischof [60] address the "kidnapped robot" problem by matching an image against a piecewise planar (indoor) scene model. They use a single keypoint match to initialize the camera pose from model-to-image matching and then compute the epipolar distance from all matches to confirm the pose. Our approach, designed for outdoor scenes, also starts from a single keypoint match, but is more robust to appearance differences and extends beyond planar surfaces.

6.1.2 Approach

A "test" image, $\mathcal{I}_t$, from the hand-held camera is matched against the world model. The model is constructed from a set of pre-aligned range scans and associated

2d intensity images, $\{\mathcal{I}_M\}$, taken by a camera which is calibrated both extrinsically and intrinsically against the range scanner. Surfaces constructed from the range scans are augmented with backprojected features from the images in $\{\mathcal{I}_M\}$. We refer to such features as "model features" and features estimated from $\mathcal{I}_t$ simply as "test features". Similarly, the backprojected keypoints extracted from images in $\{\mathcal{I}_M\}$ are referred to as "model keypoints" and the keypoints extracted from $\mathcal{I}_t$ are "test keypoints".

We follow the GDB-ICP approach from Chapter 5: the algorithm takes a hypothesis-and-test strategy and an extension of the Dual-Bootstrap algorithm [199] for refining initial transformation estimates.

The hypotheses are generated by matching model keypoints with test keypoints. Similar to the GDB-ICP 2d-to-2d image registration, a rank-ordered set of putative initial local *surface-to-image* mappings is generated. Each is considered in turn and gradually grown into a complete 3d-to-2d projection. Reminiscent of work on alignment-based recognition [92, 121], this exploits the assumption that a large fraction of the model is rigid and therefore a single model-to-image projection based on a pin-hole camera model is appropriate. Once a final projection is generated, accuracy, stability, and consistency tests are used to decide whether to accept the result, or to test the next initial mapping.

The key issues in making this strategy work are 1) how to generate the initial local mapping given the large model space and 2) how to switch effectively from the initial 2d-to-2d surface-image mapping to a 3d-model-to-2d-image projection.

The second problem, and the primary technical focus of the chapter, is how to generate a correct 3d-to-2d alignment of the model with the test image when starting from just a single keypoint match. The trick is the gradual refinement procedure of the Dual-Bootstrap approach. The model keypoints each are on a surface, so we start with a 2d-to-2d surface-to-image alignment in a small area of the model surface surrounding the keypoint. Then we gradually expand the region of the model considered and allow the algorithm to automatically switch between 2d-to-2d transformations that map a single surface to the image and the 3d-to-2d models needed for the full model-to-image projection. Model selection is

applied to automatically make these choices, eventually switching to a model that includes a full perspective camera plus one or two radial lens distortion terms. The entire refinement process, including model selection, terminates once the region is expanded to cover the region of the model (apparently) visible in the test image and the estimation converges.

The remainder of this chapter is organized as follows. Section 6.2 describes data acquisition and model construction. Section 6.3 presents the main algorithm. Section 6.4 describes a variety of experimental results. Finally, Section 6.5 summarizes our contributions and concludes the chapter.

6.2 Data, Models and Preprocessing

The 3d model is constructed from automatically-aligned 3d scans acquired using a Leica HDS 3000 LiDAR scanner. The scanner also produces the model image set, $\{\mathcal{I}_M\}$, acquired using a calibrated camera having the same optical pathway as the scanner's laser. Model images are preprocessed to extract SIFT keypoints [122], filtering the results spatially to reduce the keypoint set [25]. Keypoint locations are back-projected onto the model surfaces. Each of these "model keypoints" has an associated 3d location, scale, and surface normal. In addition, a plane π is fit to the LiDAR points in a reasonably large surface area ($80s \times 80s$, where s is the LiDAR sample spacing on the surface) surrounding the keypoint using an M-estimator. Let $\mathbf{p}_c$ be the 3d location of the model keypoint, let $\boldsymbol{\pi}_x$ and $\boldsymbol{\pi}_y$ be orthogonal axes in π, and let $\boldsymbol{\pi}_z$ be the unit normal to π. We establish a 2d coordinate system for any point $\begin{bmatrix} X & Y & Z \end{bmatrix}^\top$ on plane π by using the following transformation:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \boldsymbol{\pi}_x^\top \\ \boldsymbol{\pi}_y^\top \end{pmatrix} \left(\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - \mathbf{p}_c \right). \tag{6.1}$$

An illustration of the 2d coordinate system is shown in Figure 6.1e. This coarse surface approximation is used in the initial stage of the refinement algorithm.

Each model image is also preprocessed off-line to extract features that can be

viewed as a summary description of image content. These are face points (edge-like features) and corner points, computed at multiple scales and spread throughout the images, even in low-contrast regions. Details of this computation are provided in Section 5.3. These features are backprojected onto the range surfaces to produce "model features". The distinction between model keypoints and model features is that the model keypoints are quite sparse, have an associated 128-component descriptor vector, and are matched in the similarity space to generate initial transformations. By contrast, the model features are much more dense, more widely spread-out, and are used in the refinement and decision steps. An example section of the model with associated features is shown in Figure 1.13. Each test image, $\mathcal{I}_t$, is preprocessed in the same manner as the model images to extract keypoints and features.

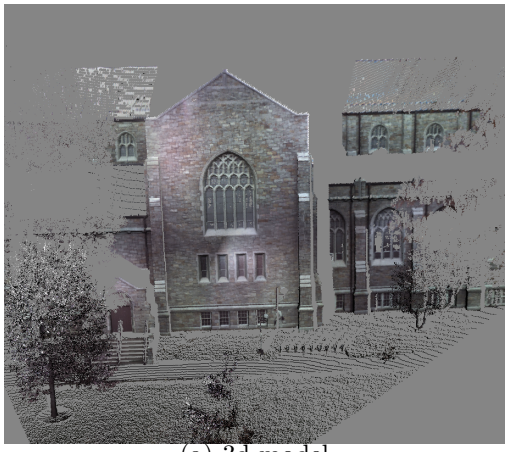
6.3 Algorithm

Mathematically, the goal is to estimate the calibration parameters of the hand-held camera, with the extrinsic parameters providing the desired pose. The estimation depends on establishing correspondence between the model features extracted from $\{\mathcal{I}_M\}$ and the test features from $\mathcal{I}_t$. The algorithm is outlined in Figure 6.2. Demonstrations of the refinement iterations are shown in Figure 6.3 and Figure 6.4.

6.3.1 Step 1: Keypoint Matching

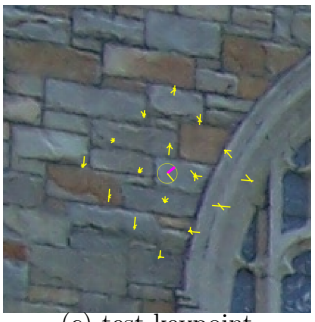
Keypoints are matched using comparison of SIFT descriptors between the test keypoints and the model keypoints. The keypoint matching process is discussed in more detail in Section 5.2. When the model is just one range scan, the keypoint matching is carried out in the same way as described in [122]. First, for each test image keypoint, $\mathbf{p}_i$, the 2 most similar model keypoints, denoted as $\mathbf{q}_j$ and $\mathbf{q}'_j$, are found from the scan. Second, the best matching model keypoint $\mathbf{q}_j$ is compared against the next best matching model keypoint $\mathbf{q}'_j$. Letting $D_{\mathbf{p}}$ be the 128-component SIFT keypoint descriptor vector, a ratio is computed for match $(\mathbf{p}_i, \mathbf{q}_j)$ as

$$r(\mathbf{p}_i, \mathbf{q}_j) = \|D_{\mathbf{p}_i} - D_{\mathbf{q}_j}\| / \|D_{\mathbf{p}_i} - D_{\mathbf{q}'_j}\|, \tag{6.2}$$



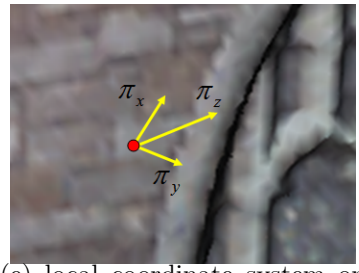
(a) 3d model

(b) test image



(c) test keypoint

(d) model keypoint



(e) local coordinate system on the plane surrounding the keypoint

Figure 6.1: An example of a keypoint match and an illustration of the local planar coordinate system. (a) shows the 3d model and (b) shows the original test image. (c) and (d) show the first correct keypoint match. Each keypoint is located at the center of the circle, with the radius of the circle representing the scale of the keypoint. The dominant gradient orientation is drawn in blue. The spread-out yellow line segments illustrate the 128-component SIFT descriptor, extracted from 4x4 histograms of intensity gradients. (e) illustrates the local coordinate system constructed around the model keypoint. π_z is the dominant gradient orientation and π_z is normal to the surface.

1. Generate rank-ordered initial keypoint matches:
   (a) For each SIFT keypoint descriptor from the camera image, $\mathcal{I}_t$, find the closest k model keypoints, under the restriction that no two model keypoints are taken from the same scan.
   (b) For each of these k matches, find the model keypoint from the same scan that is next closest to the test image keypoint descriptor and compute the ratio of descriptor distances.
   (c) Rank-order all matches for all image keypoints by increasing value of this ratio and retain the top 30.
2. For each keypoint match in rank order:
   (a) Generate an initial 2d-to-2d similarity transformation between the model keypoint's tangent plane, π, and the image plane. Initialize a small region R on π.
   (b) **Restricted 2d-to-2d Refinement:** Iterate steps of matching of features from R, re-estimation, growth of R along π, and model selection for the 2d-to-2d transformation between π and the image plane. Repeat until R reaches a minimum size.
   (c) **Full Refinement:** Continue re-estimation, region growth and refinement, now allowing consideration of 3d-to-2d camera models in addition to 2d-to-2d transformations. Growth of R is restricted to staying near π until a 3d-to-2d camera model is selected. Repeat until R covers all visible parts of the model.
   (d) Apply the three decision criteria to the resulting projection. Halt with success if all criteria pass.
3. Halt with failure if all initial transformations are rejected.

Figure 6.2: Algorithm summary.

which measures the "distinctiveness" of match $(\mathbf{p}_i, \mathbf{q}_j)$ when compared to $(\mathbf{p}_i, \mathbf{q}'_j)$. This matching and the computation of $r(\mathbf{p}_i, \mathbf{q}_j)$ are applied to all the image keypoints one by one, generating a set of keypoint matches, $\{(\mathbf{p}_i, \mathbf{q}_j) \mid \forall i\}$. Slightly different from [122], this set of keypoint matches is sorted by increasing value of $r(\mathbf{p}_i, \mathbf{q}_j)$. The 30 matches with the lowest ratios are used to generate initial model-to-image mapping estimates.

When the model is integrated from multiple scans, however, the above keypoint matching scheme is not as effective: $r(\mathbf{p}_i, \mathbf{q}_j)$ gradually loses its power of distinction as the number of model keypoints increases. Thus, we propose a modification to

the keypoint matching scheme with two differences, both designed to handle the fact that there is a large number of model keypoints across the integrated scans, to ensure that matches are spread throughout the model, and to ensure that matches need only be locally-distinct. First, for each test image keypoint, $\mathbf{p}_i$, the k best model keypoint matches are found under the restriction that no two of the matched model keypoints are from the same scan. In practice we use $k = 4$ (or the number of scans if it is smaller). Second, each of these k matches is compared against the next best matching model keypoint $\mathbf{q}'_j$, under the restriction that $\mathbf{q}'_j$ was extracted from the same scan as $\mathbf{q}_j$. (By definition $\mathbf{q}'_j$ will not be among the list of k best model keypoint matches.) The ratio $r(\mathbf{p}_i, \mathbf{q}_j)$ is computed using (6.2). The set of matches for all test image keypoints (each keypoint contributing k matches) is again sorted by increasing value of $r(\mathbf{p}_i, \mathbf{q}_j)$. As above we only take the top 30 matches to generate initial model-to-image mapping estimates.

6.3.2 Step 2a: Initial 2d-to-2d Transformation

The initial mapping is a 2d-to-2d similarity transformation between the model keypoint's approximation plane π and the image plane. The translation component aligns the model keypoint location, which is the origin on π, with the test image keypoint location. The angle between the x axis on π and the test image keypoint's gradient vector gives the rotation component. Recalling that each model keypoint has an associated, backprojected image scale, the scale parameter of the transformation is the ratio of model and test image keypoint scales.

6.3.3 Step 2b: Restricted 2d-to-2d Refinement

The restricted refinement stage is designed to extract a stable 2d-to-2d transformation between the model surface π and the image plane before allowing consideration of 3d-to-2d projections. This works even when the model keypoint is taken from a surface region that is planar over only a small area.

Model features close to π (within a few noise standard deviations) and close to the model keypoint (within $80s$, where s is the sample spacing) are projected onto the 2d coordinate system of π. These are then used by the Generalized Dual-Bootstrap algorithm (Chapter 5) to generate a 2d-to-2d transformation as though they were

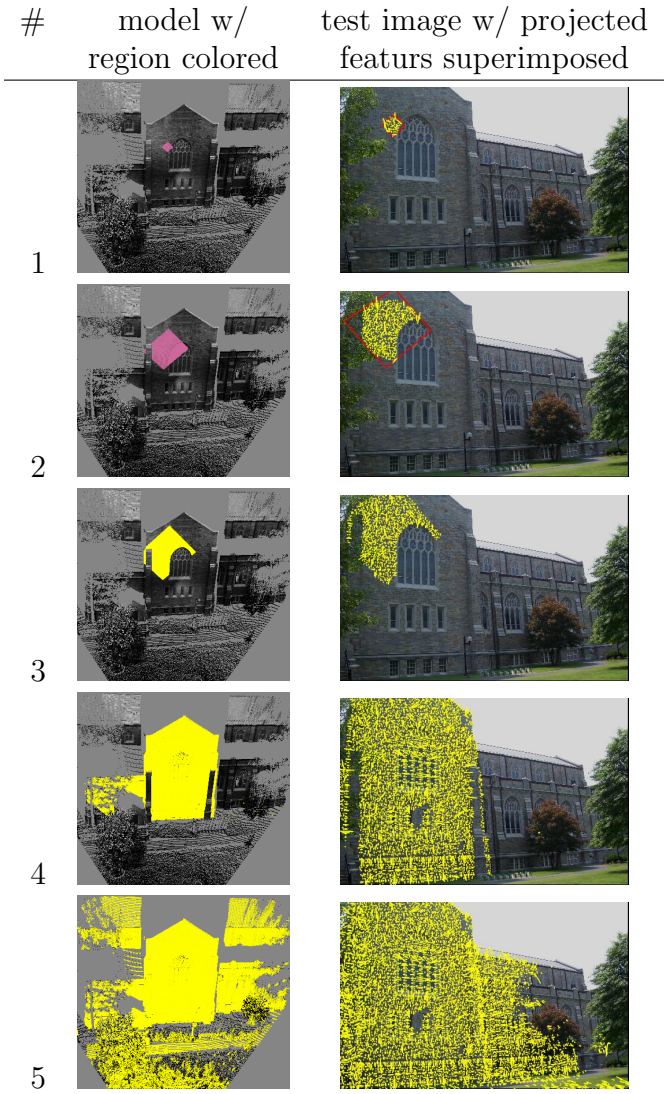


Figure 6.3: The first row shows the 3d model with the initial region on plane π superimposed (left) and the test image with the corresponding, mapped model features from within this region (right). The second row shows the last iteration of the restricted 2d-to-2d refinement before it proceeds to the full refinement. The third, fourth and fifth rows show initial, intermediate and final iterations of 3d-to-2d estimation. In the left column purple indicates 2d regions, R, on plane π, while yellow implies 3d regions R.

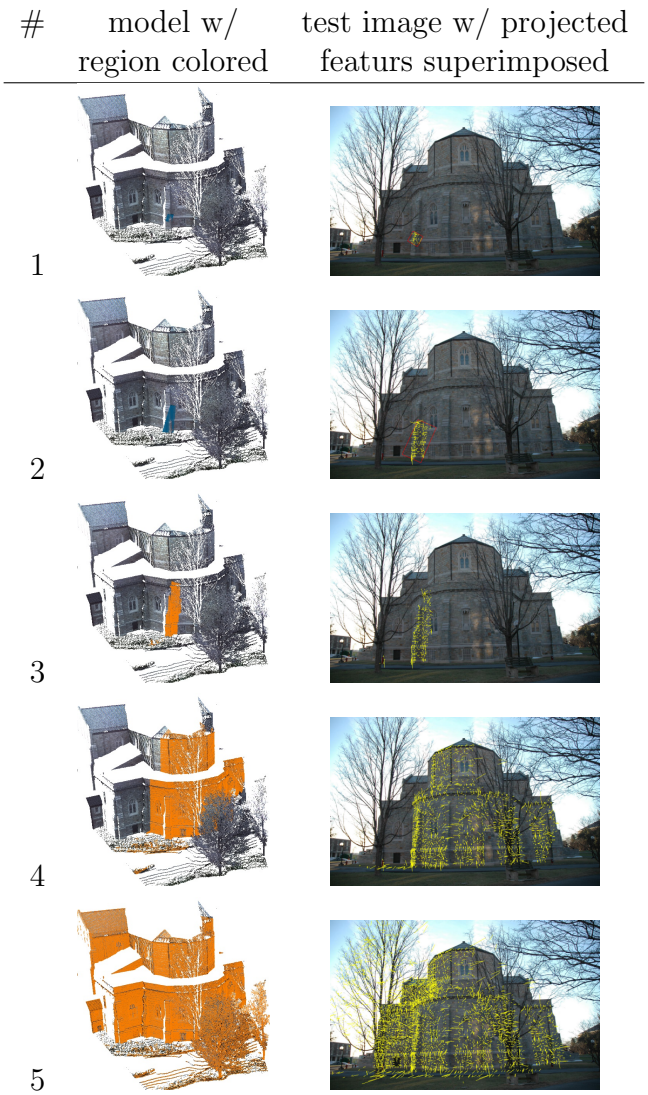


Figure 6.4:
Another illustration of the refinement process. The first row shows the 3d model with the initial region on plane π superimposed (left) and the test image with the corresponding, mapped model features from within this region (right). The second row shows the last iteration of the restricted 2d-to-2d refinement before it proceeds to the full refinement. The third, fourth and fifth rows show initial, intermediate and final iterations of 3d-to-2d estimation. In the left column purple indicates 2d regions, R, on plane π, while yellow implies 3d regions R. Notice that in this case the algorithm starts from a curved surface that is not well-approximated by a plane. Also, at the end of the restricted refined (the 2nd row) and the beginning of the full refinement, the regions become elongated in one direction, demonstrating how the region grows faster along directions where the estimate is more stable.

simply image-plane features. This allows symmetric model-to-image and image-to-model matching, with both sets of matches used in estimating the transformation parameters. This prevents singularities that can occur early in the registration process, especially when the scaling is unstable. It was shown in Section 5.6 that, when aligning challenging 2d image pairs, not using symmetric matching causes the loss of 28% of the successful keypoint initializations and three of the most difficult pairs, including one pair with significant scale difference.

During this restricted refinement, the bootstrap region R is an axis-aligned rectangle on plane π. R is initialized as a square with half-width $3\sigma + 20s$, where σ is the model keypoint scale and s is the LiDAR sample spacing.

6.3.3.1 Matching and Estimation

Re-using the notation from Chapter 5, the mapping function is denoted as $\mathbf{T}(\mathbf{p};\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the parameter vector. Notice that when the current model is a 2d-to-2d mapping, $\mathbf{T}(\mathbf{p};\boldsymbol{\theta}) : \mathcal{R}^2 \rightarrow \mathcal{R}^2$. When the current model is a 3d-to-2d projection, $\mathbf{T}(\mathbf{p};\boldsymbol{\theta}) : \mathcal{R}^3 \rightarrow \mathcal{R}^2$. Given an estimate $\hat{\boldsymbol{\theta}}$ and a set $\mathcal{P}$ of model features sampled from R, for each $\mathbf{p}_i \in \mathcal{P}$, the test-image feature point $\mathbf{q}_i$ closest to $\mathbf{T}(\mathbf{p};\boldsymbol{\theta})$ is located, and the pair $(\mathbf{p}_i, \mathbf{q}_i)$ is added as a correspondence. This is done for both the corner points and the face points, each feature being matched to features of the same type. The result is two correspondence sets, denoted $\mathcal{C}_c$ for corners and $\mathcal{C}_f$ for face points. The transformation parameters are then re-estimated by minimizing

$$\begin{aligned} E(\boldsymbol{\theta}) = & \sum_{(\mathbf{p}_i,\mathbf{q}_i)\in\mathcal{C}_e} \rho((\mathbf{T}(\mathbf{p}_i;\boldsymbol{\theta}) - \mathbf{q}_i)^\top \boldsymbol{\eta}_i/\sigma_e) \\ & + \sum_{(\mathbf{p}_i,\mathbf{q}_i)\in\mathcal{C}_c} \rho(\|\mathbf{T}(\mathbf{p}_i;\boldsymbol{\theta}) - \mathbf{q}_i\|/\sigma_c). \end{aligned} \tag{6.3}$$

Here, ρ is the Beaton-Tukey biweight robust loss function (see [200]), and σ_c and σ_e are the robustly-estimated standard deviations of alignment errors for the corner and face points separately. Different error norms and standard deviations are used because the two feature types have different distance measures and error properties. A more detailed discussion about the error norms can be found in Section 5.4.3.

Iteratively-reweighted least-squares is used to minimize (6.3) [200]. This is

combined with the Levenberg-Marquardt optimization technique [113, 132] for models more complex than affine.

Note that Step (2c) also uses a matching and estimation process similar to what is used here (see below). The difference is that in the restricted estimation in Step (2b) the roles of image features and model features are reversed in matching, producing two more sets of correspondences in equation (6.3). In Step (2c) this is not used, in part for efficiency and in part because of the ambiguity of 2d to 3d matching.

6.3.3.2 Region Growth and Model Selection

Once the parameters are estimated, the covariance matrix of the estimate is obtained using the inverse of the Hessian of (6.3) evaluated at the parameter estimate $\hat{\boldsymbol{\theta}}$. This is used to control growth of R parallel to π, with more certainty in the estimate leading to faster growth. The details of the model selection were presented in Section 5.4.4. As a reminder, the model selection step uses the modified form of the Aikaike Information Criterion [27] as in :

$$\begin{aligned}\mathcal{I} = &-2\left[\mid \mathcal{C}_c \mid \log(\sigma_c) + \mid \mathcal{C}_f \mid \log(\sigma_f) + E(\hat{\boldsymbol{\theta}}; \mathcal{C}_c, \mathcal{C}_f)\right] \\ &+ 2nl/(n-l-1),\end{aligned} \tag{6.4}$$

where l is the degrees of freedom in current model, $n = 2 \mid \mathcal{C}_c \mid + \mid \mathcal{C}_f \mid$ is the effective number of constraints (each corner match provides with two constraints, while each face point match provides one), and the term $2nl/(n-l-1)$ adjusts for small-sample bias [27, p. 51].

Here, four 2d-to-2d models are used: similarity, affine, plane homography, and plane homography plus radial lens distortion. Finally, region growth, and therefore all of Step (2b), terminate when expansion of R includes all of the selected points. This means R is large enough and therefore the mapping is stable enough to consider switching to 3d-to-2d models.

6.3.4 Step (2c): Full Refinement

Step (2c) uses a similar matching and estimation process to the one in Step (2b) (see Section 6.3.3.1). Here we focus on the transition from 2d-to-2d transformation to 3d-to-2d camera models using the model selection technique and the details of region growth.

6.3.4.1 The Transition

The transition from 2d-to-2d mapping to 3d-to-2d camera projection is essentially a camera calibration problem using a planar surface [204, 224, 241]. (For more complete reviews on camera calibration methods, please refer to Section 2.3.3.)

Taking Zhang's and Sturm's approach [204, 241], we compute a simplified camera by using the homography matrix and the orthogonal constraints. First, we apply a rigid transformation similar to the one shown in Equation 6.1 to transform points on the plane π to the new plane $z = 0$, such that the points have new coordinate $\begin{bmatrix} x & y & 0 \end{bmatrix}^\top$. The desired camera projection in homogeneous coordinates is

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix},$$

where $\mathbf{K}$ is the camera calibration matrix and $\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix}$ is the rotation matrix and $\mathbf{t}$ is the translation. Notice it can be simplified to:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

Then, if $\mathbf{H}$ is the 3×3 matrix representing the estimated 2d-to-2d plane projective

transformation, the mapping of point $\begin{bmatrix} x & y \end{bmatrix}^\top$ on plane π into the image plane is

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \mathbf{H} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix},$$

which implies $\mathbf{H} = s\mathbf{K} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix}$ up to some unknown non-zero scalar s. Letting $\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \end{bmatrix}$, the equation can be written as

$$\begin{aligned} \mathbf{h}_1 &= s\mathbf{K}\mathbf{r}_1 \\ \mathbf{h}_2 &= s\mathbf{K}\mathbf{r}_2 \\ \mathbf{h}_3 &= s\mathbf{K}\mathbf{t}, \end{aligned}$$

or equivalently

$$\begin{aligned} \mathbf{r}_1 &= \mathbf{K}^{-1}\mathbf{h}_1/s \\ \mathbf{r}_2 &= \mathbf{K}^{-1}\mathbf{h}_2/s \\ \mathbf{t} &= \mathbf{K}^{-1}\mathbf{h}_3/s. \end{aligned} \tag{6.5}$$

Recall that rotation matrix $\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix}$ is orthonormal

$$\mathbf{r}_1^\top \mathbf{r}_2 = 0, \qquad \mathbf{r}_1^\top \mathbf{r}_1 = 1, \qquad \mathbf{r}_2^\top \mathbf{r}_2 = 1.$$

This constitutes orthogonality constraints for the above equation:

$$\mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 = 0, \tag{6.6}$$

$$\mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_1 = \mathbf{h}_2^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2. \tag{6.7}$$

Now assuming the skew of the camera is 0 and the principal point $\begin{bmatrix} u_0 & v_0 \end{bmatrix}^\top$ is fixed at the center of the image, we have a simplified calibration matrix with only

two unknowns α_x and α_y:

$$\mathbf{K} = \begin{bmatrix} \alpha_x & 0 & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Substituting the above into Equation 6.6 and Equation 6.7, we can solve for the two unknowns α_x and α_y, followed by the computation of the extrinsic parameters using Equation 6.5.

The simplified camera has two intrinsic and six extrinsic parameters, accounting for the eight degree of freedom of the plane homography. Thus, a unique solution can be computed in general using the above formulation. It is worth mentioning that Sturm and Maybank enumerated all critical configurations when using this approach [204]. In our settings, our method is degenerate only when the plane π is parallel to the image plane.

Alternative methods for camera calibration using a single planar target can be found in [38, 224, 234].

After initialization, the camera estimate is refined using the Levenberg-Marquardt (L-M) optimization technique [113, 132]. The final camera estimate, along with the correspondence sets $\mathcal{C}_f$ and $\mathcal{C}_c$, is passed to the model selection criteria to decide when a 3d-to-2d camera projection is more suitable than a 2d-to-2d mapping.

6.3.4.2 Model Selection

In full refinement, we expand R to a volume by adding a component, π_z, normal to the planar surface of π. Initially, the width of R normal to π is 10 standard deviations (obtaining using the robust estimation of the parameters of π), large enough to include some points as the surface curves or crosses a crease boundary. The rectangular axes of R remain aligned with the coordinate system of π throughout the computation (for this initial estimate).

It is important to consider the challenge here. When using planar camera calibration techniques [204, 224, 241], the estimates tend to be unstable, however, especially for smaller planes and for projecting 3d points far from the planes. For our problem, this affects both camera location estimate and the decision criteria.

More specifically, our growth and refinement process only works effectively if, when R expands, the newly-included points can be matched reliably using the estimated projection parameters. This fails when the camera matrix is too unstable because matches for the new points are likely to be incorrect, driving the transformation estimate in the wrong direction. On the other hand, if we rely on a planar region for too long, then the planar approximation will be inaccurate, leading again to incorrect matching and estimation.

We address this using model selection, allowing competition between 2d-to-2d transformation models and 3d-to-2d projection models. We use the modified form of the Aikaike Information Criteria, same as Equation 6.4. It generally trades-off the stability of lower-order models and the accuracy of higher-order models. In our case, when R encloses points that are only from a planar surface, model selection should tend to choose a 2d-to-2d transformation, at least until a large-enough set of features is included in R (Figure 6.3 and Figure 6.4). When points from a different surface (e.g. at a boundary) are included in R, or when the surface starts to curve substantially, a 3d-to-2d model will appear more stable earlier in the computation, and the algorithm will choose it.

Thus, during Step (2c) points in R are used to estimate both a 2d-to-2d transformation and a 3d-to-2d camera projection until the algorithm selects a 3d-to-2d projection (after R has expanded sufficiently). Once the algorithm switches to the 3d-to-2d projection, 2d-to-2d transformations are no longer considered. Prior to this, when the chosen model is 2d-to-2d, matching of model features in R occurs by projecting the points onto π and then, using the estimated 2d-to-2d transformation, onto the image plane. The closest test image feature is then found. This generates one element of the correspondence set for each feature.

The parameters of all transformations currently under consideration are estimated using the same set of correspondences. The covariance matrices of these estimates are computed, and model selection is applied. For 2d-to-2d transformations, the four models described above are used, while for 3d-to-2d transformations four additional models are used —- the models are listed in Table 6.1. Once a switch is made to a higher order model, the algorithm does not switch back, so fewer than

eight models are typically considered during any one iteration.

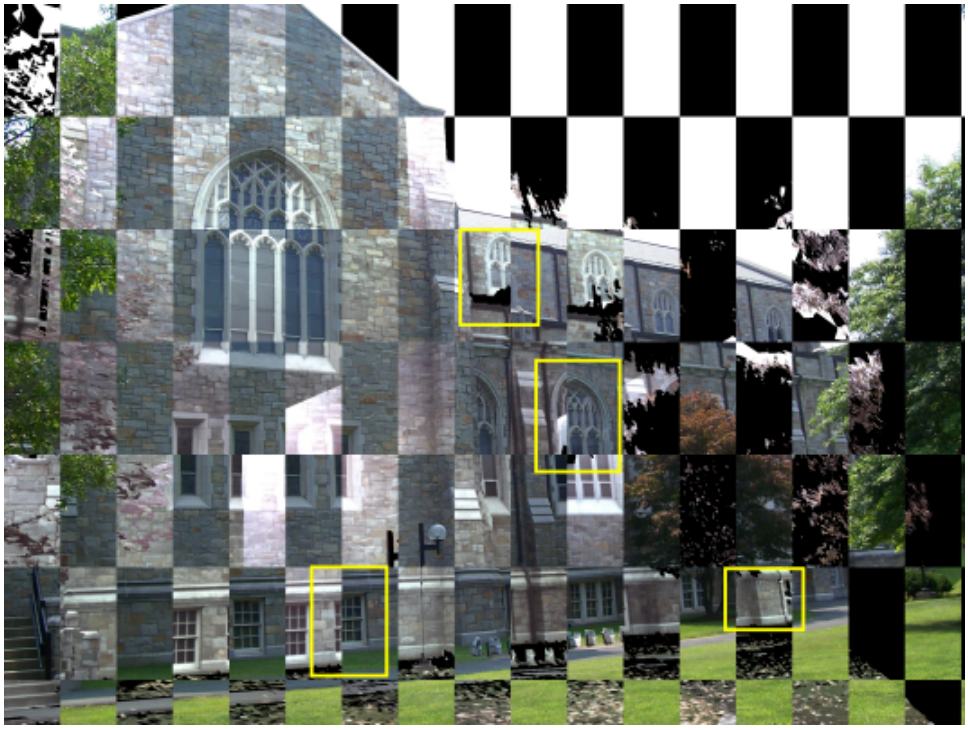
Figure 6.5 illustrates the accuracy the estimate of the first selected camera model, which is the 11-degree-of-freedom perspective camera. Figure 6.5a shows a checkerboard image between the test image $\mathcal{I}_t$ and a "projection image", which is generated by projecting the 3d model through the estimated camera. Although the dominant plane on the left of the image is well aligned, many other regions, including those outlined in yellow box, are mis-aligned. This indicates this camera estimate is only accurate in the current region (the region is shown in the left image of the 3rd row in Figure 6.3). As the region grows and more constraints are included in the estimation, the camera estimate will be more and more accurate.

Transformation	mapping on plane π?	number of radial coefficients terms	degrees of freedom
Similarity	Yes	0	4
Affine	Yes	0	6
Homography	Yes	0	8
Homography	Yes	1	9
Reduced camera	No	0	8
Perspective camera	No	0	11
Perspective camera	No	1	12
Perspective camera	No	2	13

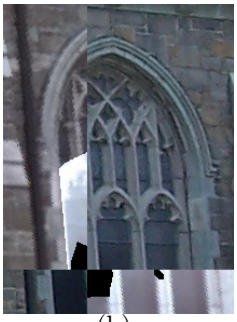
Table 6.1: The suite of models use during the full estimation. Note that model switching is monotonic — it can only switch to higher order models than the current one. Also, the switching can pick any of the higher order models, not necessarily the next higher one.

6.3.4.3 Region Growth

The region growth depends on whether a 2d-to-2d model or a 3d-to-2d model is selected. In the former case, the model is expanded only along π_x and π_y (shown in Figure 6.6a). In the latter case, expansion is allowed along π_z, normal to π, as well (shown in Figure 6.6b. In either case, growth is controlled by the uncertainty in the mapping of points centered on each face of R (four sides for growth in the plane only). To measure this uncertainty, let $\hat{\mathbf{\Sigma}}_{\boldsymbol{\theta}}$ be the parameter estimate covariance matrix and let the Jacobian of the transformation be $\mathbf{J} = \frac{\partial \mathbf{T}}{\partial \boldsymbol{\theta}}\left(\mathbf{p}; \hat{\boldsymbol{\theta}}\right)$. The covariance matrix of the mapped point of $\mathbf{p}$ (in the image) — or forward transfer error of $\mathbf{p}$ —



(a) Checkerboard image



(b)

(c)

(d)

(e)

Figure 6.5: (a) shows a checkerboard image between the original image and the projection image generated by projecting the 3d model through the estimated camera (same iteration as the 3rd row in Figure 6.3). Notice even though the plane on the left is well aligned, many other regions, including those outlined in yellow, are mis-aligned. The right half of the window in (b) is completely missing. The windows in (c) and (d) are mis-aligned between the adjacent two checkers. The base block of the building in (e) is mis-placed on the grassland. This camera estimate will be further refined until the iteration converges. The final accurate camera estimate is shown in Figure 6.7.

is

$$\mathbf{\Sigma_p} = \mathbf{J}\hat{\mathbf{\Sigma}}_{\boldsymbol{\theta}}\mathbf{J}^\top. \tag{6.8}$$

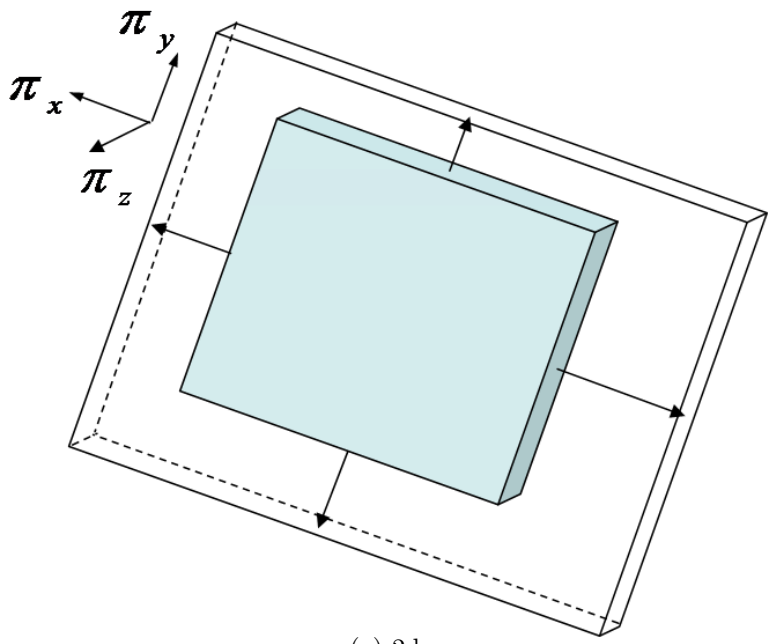
The face (side in 2d) of R is expanded outward in inverse proportion to the trace of this "transfer-error" covariance matrix. When the algorithm has switched to a 3d-to-2d mapping, the uncertainty in the mapping tends to cause slower growth normal to π than tangent to π.

When plane π is approximately parallel to the image plane (the degenerate configuration), the camera estimate obtained using Zhang's and Sturm's method suffers from numerical instability. Points far off plane π are likely to be projected to incorrect locations. Thus, matching points far off plane π may result in incorrect camera estimate. On the other hand, matching points close to the plane is safe, because the numerically-degenerate camera estimate is still accurate for the projection of points on the plane. This is captured in the computation of $\mathbf{\Sigma_p}$ in (6.8) — when the configuration is degenerate, the forward transfer error $\mathbf{\Sigma_p}$ indicates it is unstable to grow along $\boldsymbol{\pi}_z$, the normal direction. Therefore, the region will grow slowly along $\boldsymbol{\pi}_z$, allowing correct matching of points close to the plane, until there are enough constraints for the camera estimation. Thus, the algorithm naturally handles this (approximately) degenerate configuration through a combination of the model selection and the region growth described above, unless plane π is also "isolated" from the rest of the 3d model, which will be discussed later.

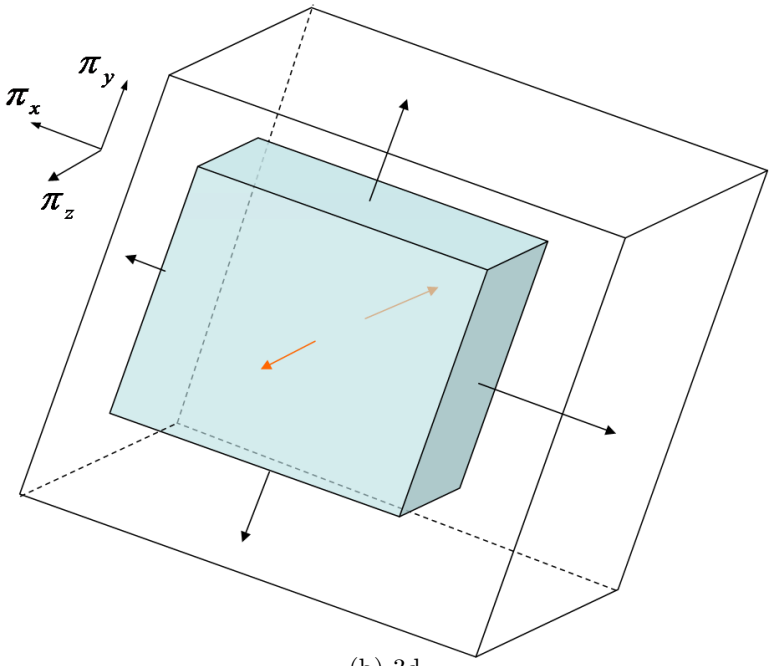
6.3.5 Step (2d): Decision Criteria

Step (2d) terminates when R covers the field of view of the model from the (estimated) perspective of test image I_t. In this case, the algorithm evaluates the resulting 3d-to-2d projection using the decision criteria. If these all pass, the algorithm halts with success.

The decision criteria are straightforward adaptations from the Generalized Dual-Bootstrap algorithm (Chapter 5). The first is a threshold on the robustly estimated distance between projected model points and their corresponding image points. The second is the stability in the transformation, measured by the trace of the transfer error covariance matrix (6.8) on the boundaries of the region. Poorly



(a) 2d



(b) 3d

Figure 6.6: Illustration of the region growth. When the current model is a 2d-to-2d mapping, the region grows along π_x and π_y (shown in (a)). When the current model is a 3d-to-2d projection, the region grows along all three directions, including π_z, normal to π (shown in (b)).

constrained estimates (which are likely to be based on incorrect correspondences) produce transfer error covariances with relatively large trace values. The third criterion measures the consistency in the constraints by measuring the distribution of the angles between mapped model features — with directions mapped into the image plane of $\mathcal{I}_t$ — and their corresponding image features. For correct mappings the angles tend to be clustered near 0. For challenging cases, especially involving substantial changes in viewpoint, illumination, or even scene content, sometimes incorrect mappings fail only one of the three criteria. This makes all criteria necessary. As a final comment, the decision criteria are also applied during refinement, with higher tolerances, to quickly eliminate estimates that started from incorrect matches.

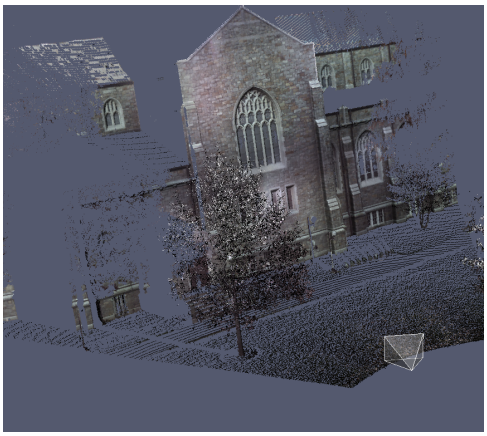
6.4 Experiments

We present experimental evidence showing the effectiveness of our proposed algorithm. Scans were taken across several overlapping areas of campus, covering approximately a 100m x 100m region. Nine scans, each with a large field of view, were collected and integrated. Together these scans contain 55,131 model keypoints. Sixty test images were collected from within the same area. These were taken weeks or months later than the scans, including seven at night, 17 during the winter with snow on the buildings and the ground, and 15 from the same viewpoint but with varying focal lengths. We use these scans and test images to evaluate our algorithm.

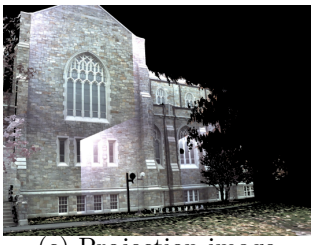
6.4.1 Overall

The first result is simply an evaluation of how many of the test images were "correctly" located. We judge correctness here by using the estimated 3d-to-2d transformation (camera) parameters to create a synthetic image from the model and visually compare this image against the test image. If the location is correctly determined, the two images should be very similar, except for illumination differences. The images in Figure 6.8, Figure 6.9, Figure 6.10, and Figure 6.11 show checkerboard mosaics created by extracting alternate blocks of the synthetic and test images.

(a) Original image



(b) The camera location and orientation



(c) Projection image

(d) Checkerboard image

Figure 6.7: The final camera projection estimate obtained from image (a), after decomposition, gives the camera location and orientation (shown in (b) as a view frustum superimposed with the model). (c) shows the projection image by projecting the 3d model through the estimated camera. (d) shows a checkerboard mosaic of the two images, showing no misalignments and therefore indicating the accuracy of the estimate.

Figure 6.8: Result on the night image shown in Figure 1.13d. The upper left shows the test image and the upper right shows the projection image. The bottom shows a checkerboard mosaic of the two images, showing no misalignments and therefore indicating the accuracy of the estimate. Keypoint matching together with a RANSAC search failed on this model.

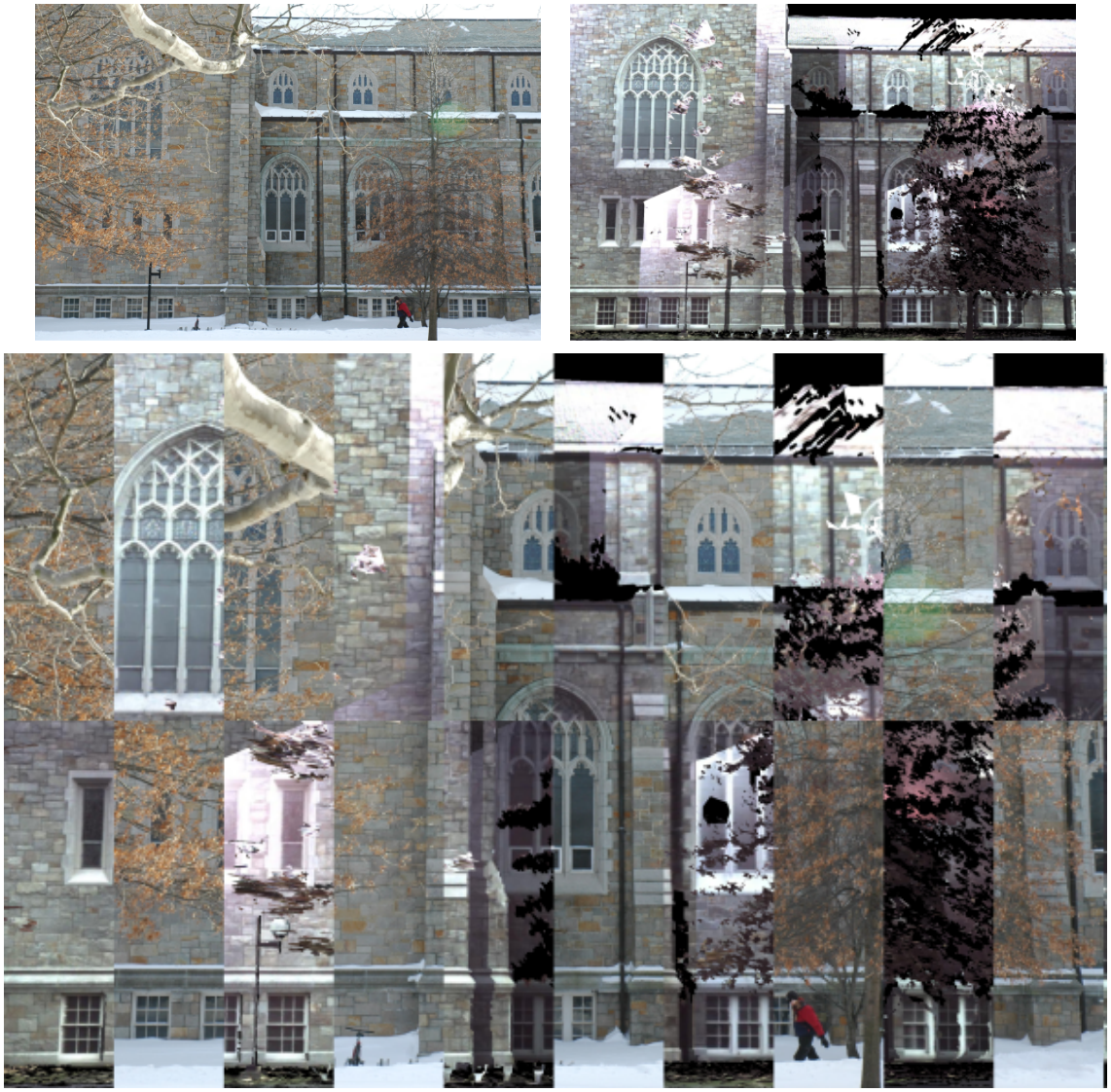


Figure 6.9: Result on the winter snow image shown in Figure 1.13e. The upper left shows the test image and the upper right shows the projection image. The bottom shows a checkerboard mosaic of the two images, showing no misalignments and therefore indicating the accuracy of the estimate. Keypoint matching together with a RANSAC search failed on this model.

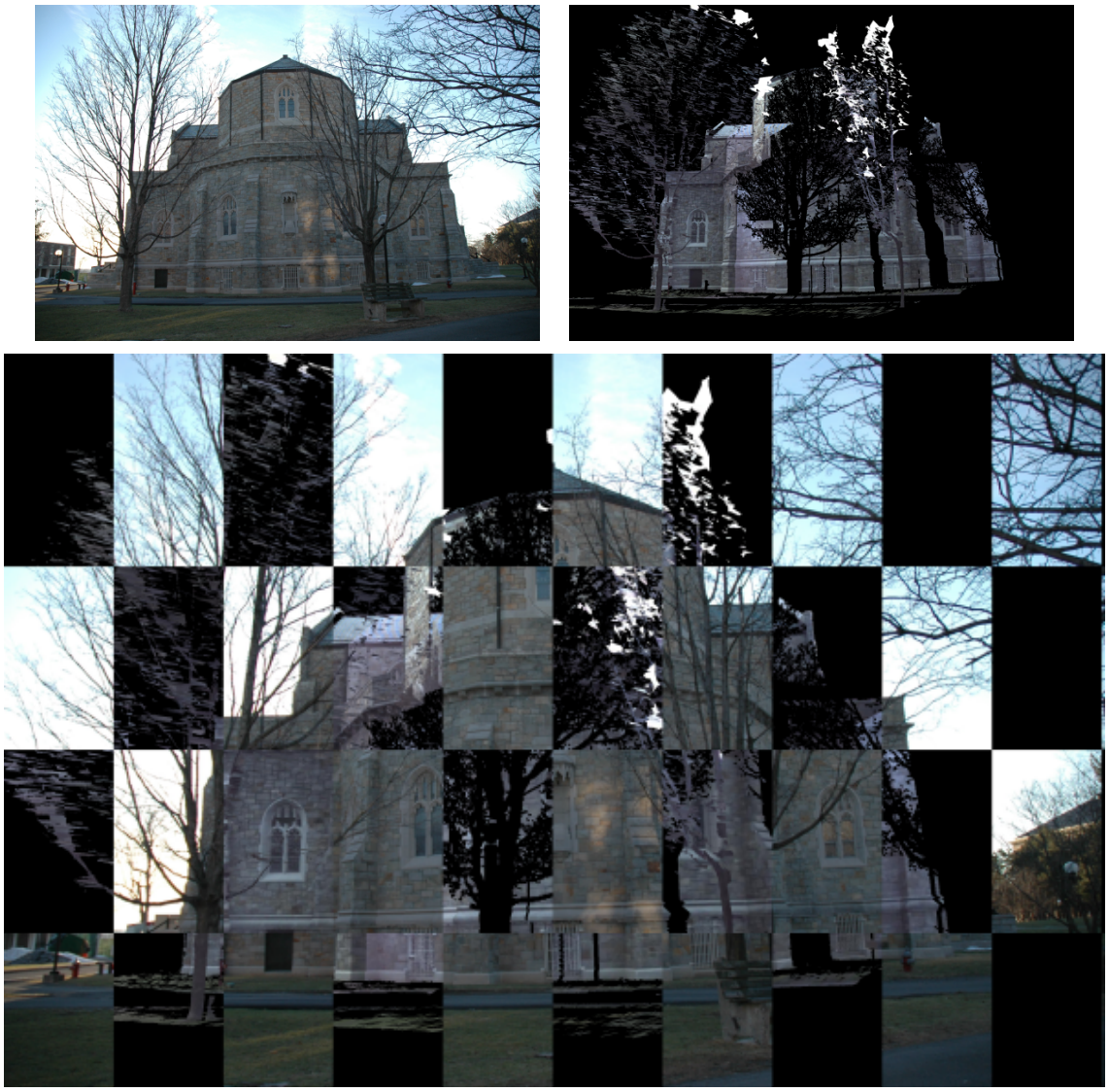


Figure 6.10: Result on a test image involving a model region with small planar surfaces (the refinement process is illustrated in Figure 6.4). The upper left shows the test image. The upper right shows the projection image. The dark shadow of the tree represents a hole in the model where data are unavailable due to occlusions. The bottom shows a checkerboard mosaic of the two images, showing no misalignments and therefore indicating the accuracy of the estimate. Keypoint matching together with a RANSAC search failed on this model.

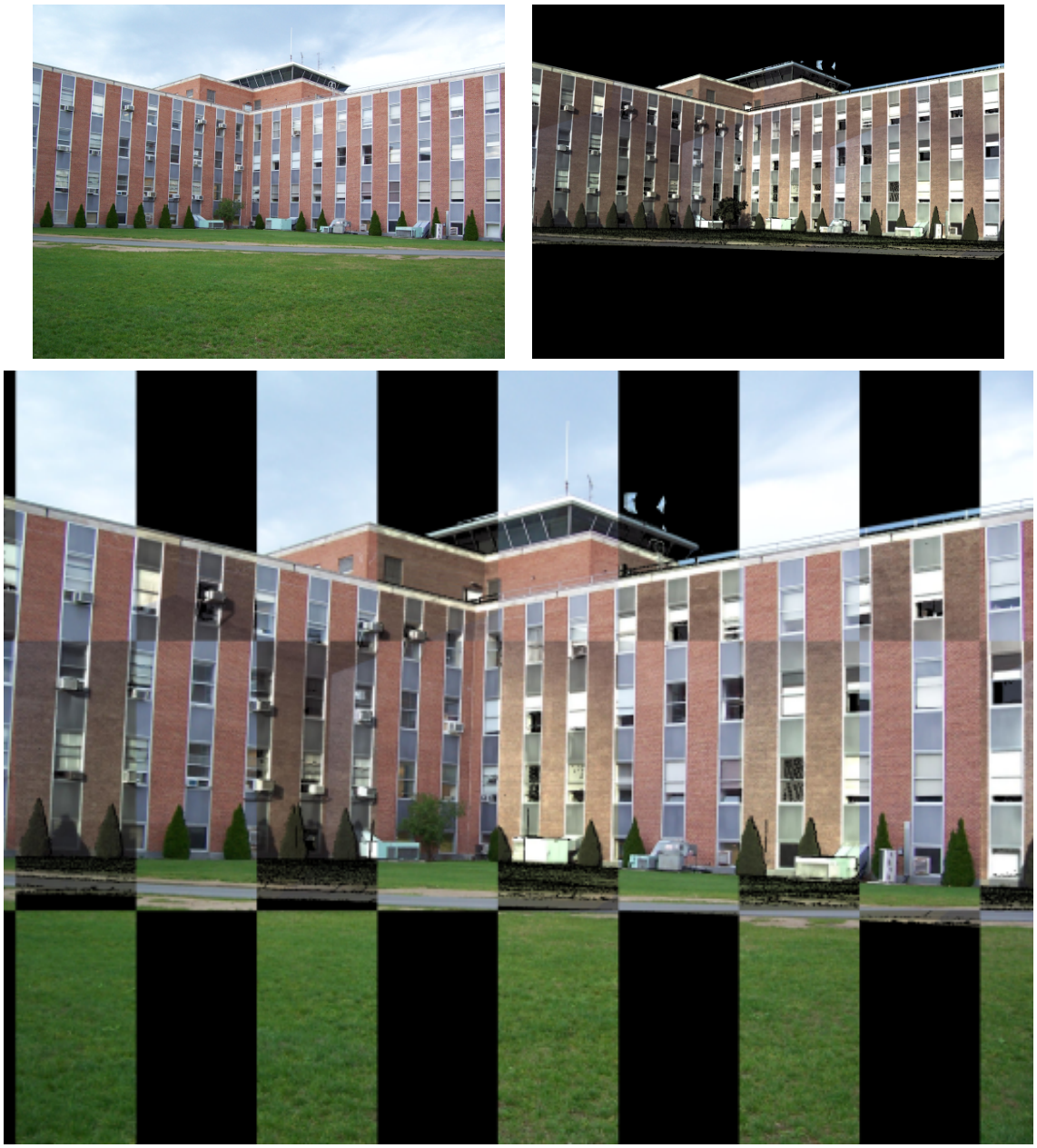


Figure 6.11: Example result for a test image from a part of the model containing a repetitive building structure, with test image, projection image and checkerboard all shown.

Of the 60 test images, the algorithm automatically and correctly estimated the camera location of 52. For the remaining 8, the algorithm indicated that it could not find an alignment. The correct alignments include images with different illumination (Figure 6.8), physical changes (Figure 6.9), with significant amount of occlusion (Figure 6.10), and with different scalings (Figure 6.13a and Figure 6.13c). A close examination revealed that the 8 failures are due to poor image contrast, low overlap with the model, or dramatic scaling changes. The failures will be discussed in Section 6.4.5.

6.4.2 Accuracy and Stability of the Estimates

To quantitatively measure the accuracy of our estimated camera locations, we conducted the following experiment using two overlapping 3d models and three intensity images from $\{\mathcal{I}_M\}$ of each of the two models (shown in Figure 6.12). Denote the two models as Model A and Model B. The ground-truth 3d rigid transformation between Model A and Model B is computed using a new 3d registration method [103], followed by manual verification. We have mentioned above (in Section 6.1.2) that the intensity images, $\{\mathcal{I}_M^{A1}, \mathcal{I}_M^{A2}, \ldots\}$, used in the construction of Model A, were taken from a camera which is calibrated extrinsically and intrinsically against the range scanner. Hence, for the three images, $\mathcal{I}_M^{A1}$, $\mathcal{I}_M^{A2}$, and $\mathcal{I}_M^{A3}$, their camera locations with respect to Model A are known a priori. The same applies to the camera locations of $\mathcal{I}_M^{B1}$, $\mathcal{I}_M^{B2}$, and $\mathcal{I}_M^{B3}$ with respect to Model B. Using the ground-truth 3d rigid transformation, we obtain the ground-truth camera location of images $\mathcal{I}_M^{A1}$, $\mathcal{I}_M^{A2}$, and $\mathcal{I}_M^{A3}$ with respect to Model B. In comparison, we also obtain the camera location estimates of these images with respect to Model B by applying our algorithm. The ground-truth and estimated values of the camera location and the camera intrinsic parameters are listed in Table 6.2[14]. The same comparison is conducted between the Model A and images $\mathcal{I}_M^{B1}$, $\mathcal{I}_M^{B2}$, and $\mathcal{I}_M^{B3}$. The results are listed in Table 6.2.

Based on the Euclidean distance between the ground-truth and estimated camera locations, the best camera location estimates are from images $\mathcal{I}_M^{A1}$ and $\mathcal{I}_M^{A3}$, where

[14] It is important to point out that this comparison is valid because the images, $\mathcal{I}_M^{A1}$, $\mathcal{I}_M^{A2}$, and $\mathcal{I}_M^{A3}$, are *NOT* used in the construction of Model B, and vice versa.

With Model B	Ground Truth	$\mathcal{I}_M^{A1}$ Est.	$\mathcal{I}_M^{A2}$ Est.	$\mathcal{I}_M^{A3}$ Est.	$\lvert\Delta\rvert$ Avg.	$\lvert\Delta\rvert$ Min	$\lvert\Delta\rvert$ Max
α_x	2347.5	2311.9	2303.4	2347.7	26.6	0.20	44.1
α_y	2345.7	2310.2	2289.5	2346.4	30.8	0.70	56.2
skew k	0	-3.6	-7.6	-2.4	4.53	2.44	7.57
u_0	515.5	499.9	469.1	505.6	23.9	9.87	46.4
v_0	524.9	515.3	531.8	525.6	5.75	0.67	9.64
x (in m.)	-11.69	-11.25	-10.81	-11.70	0.44	0.01	0.87
y (in m.)	14.71	14.73	14.81	14.62	0.07	0.02	0.10
z (in m.)	0.03	-0.02	-0.02	-0.01	0.12	0.05	0.21
dist. (in m.)	0	0.44	0.90	0.14	0.50	0.14	0.90
With Model A	**Ground Truth**	**$\mathcal{I}_M^{B1}$ Est.**	**$\mathcal{I}_M^{B2}$ Est.**	**$\mathcal{I}_M^{B3}$ Est.**	**$\lvert\Delta\rvert$ Avg.**	**$\lvert\Delta\rvert$ Min**	**$\lvert\Delta\rvert$ Max**
α_x	2347.5	2204.6	2302.5	2421.3	87.2	44.9	142.9
α_y	2345.7	2287.7	2296.9	2404.6	55.2	48.7	58.9
skew k	0	0.38	-5.9	13.84	6.70	0.38	13.84
u_0	515.5	475.6	488.0	523.4	25.1	7.90	13.84
v_0	524.9	592.1	531.1	412.1	62.07	6.21	112.8
x (in m.)	-13.16	-13.37	-13.12	-12.05	0.45	0.04	1.11
y (in m.)	13.41	11.14	12.76	13.94	1.15	0.53	2.28
z (in m.)	0.19	0.84	0.10	-0.24	0.39	0.08	0.65
dist. (in m.)	0	2.38	0.66	1.30	1.45	0.66	2.38

Table 6.2: Comparison between the ground-truth and estimated camera location and intrinsics. The rows marked with x, y, and z are the corresponding coordinate of the camera locations. The rows marked with "dist." represent the Euclidean distance from the estimated camera locations to the ground truth. $|\Delta|$ represents the absolute difference between the estimate and the ground-truth value.

the distance is less than 0.5 meters. The worst estimates are from images $\mathcal{I}_M^{B1}$ and $\mathcal{I}_M^{B3}$. This is partly due to the dominant plane and the occluded structures, the combination of which results in small variation in depth. In general, considering the camera is 18.8 meters away from the scanner, all of these camera location estimates are quite accurate, having a relative error range between 0.7% to 12%.

We present two ways to quantitatively measure the stability of our estimated camera locations and make a comparison with the results from a keypoint-based method. This keypoint-based method uses Lowe's keypoint extraction and matching [122] followed by RANSAC estimation and L-M optimization after collecting all the inliers. First, we took 11 test images from the same location while varying the

(a) Model A

(b) Model B

(c) $\mathcal{I}_M^{A1}$

(d) $\mathcal{I}_M^{B1}$

(e) $\mathcal{I}_M^{A2}$

(f) $\mathcal{I}_M^{B2}$

(g) $\mathcal{I}_M^{A3}$

(h) $\mathcal{I}_M^{B3}$

Figure 6.12: The two models and three of the associated intensity images from each model are shown in two columns. Using these models, images, and the ground-truth 3d rigid transformation between the two models, we can evaluate the accuracy of our algorithm.

focal length of the test image camera from 18mm to 70mm (Figure 6.13). We ran both algorithms on each test image separately and plotted the measured camera location parameters in a coordinate system centered at the origin of the scanner from the closest model scan. Results indicating the differences of the two algorithms are shown in Figure 6.14. Our algorithm obtained more stable estimates than the keypoint-based method. Moreover, the keypoint-based method failed to produce reasonable results on the images beyond 55mm focal length.

In the second experiment, using two cameras we took two separate sequences, each with 3-meter steps between test images (Figure 6.15). We applied both the algorithms to each image separately. We then measured the distance between camera locations for adjacent frames in each sequence. The resulting values, which should each be 3 meters, are plotted in Figure 6.16. The only substantial error for our algorithm is $\mathcal{I}_9$ (shown in Figure 6.15), where the test camera entered an area substantially occluded in the 3d model. Otherwise, the relative locations are quite reliable. As a final comment on the results, the keypoint-based method failed to produce reasonable results on five images — $\mathcal{I}_7$ to $\mathcal{I}_{10}$ and $\mathcal{I}_{19}$ — all of which have part of the building facade being occluded.

6.4.3 Keypoint Statistics

We can study the reasons for the failure of using keypoint matching and then RANSAC to estimate the cameras by considering the number and fraction of correct keypoint matches generated in the initialization phase of our algorithm. For example for the image shown in Figure 6.10, there are total 63 keypoint matches that have the ratio (Equation 6.2) smaller than the threshold 0.8 — the threshold used by the SIFT matching algorithm [122]). (The top 30 of these are tested by our algorithm.) Only 5 of these are correct, which we judge automatically based on consistency with a manually-validated camera model. The cause of this is the substantial amount of occlusion. For the after-snow image in Figure 6.9, 69 keypoint matches passed the 0.8 ratio threshold, but only 10 are correct. For the night image in Figure 6.8, only 13 out of the 67 matches are correct. While our algorithm succeeds on these images, having so few correct keypoint matches, both in terms of actual numbers


(a) focal length 18mm


(b) focal length 55mm


(c) focal length 70mm


(d) 3d model

Figure 6.13: Samples of the test images taken from one viewpoint but at varying focal lengths and the 3d model.

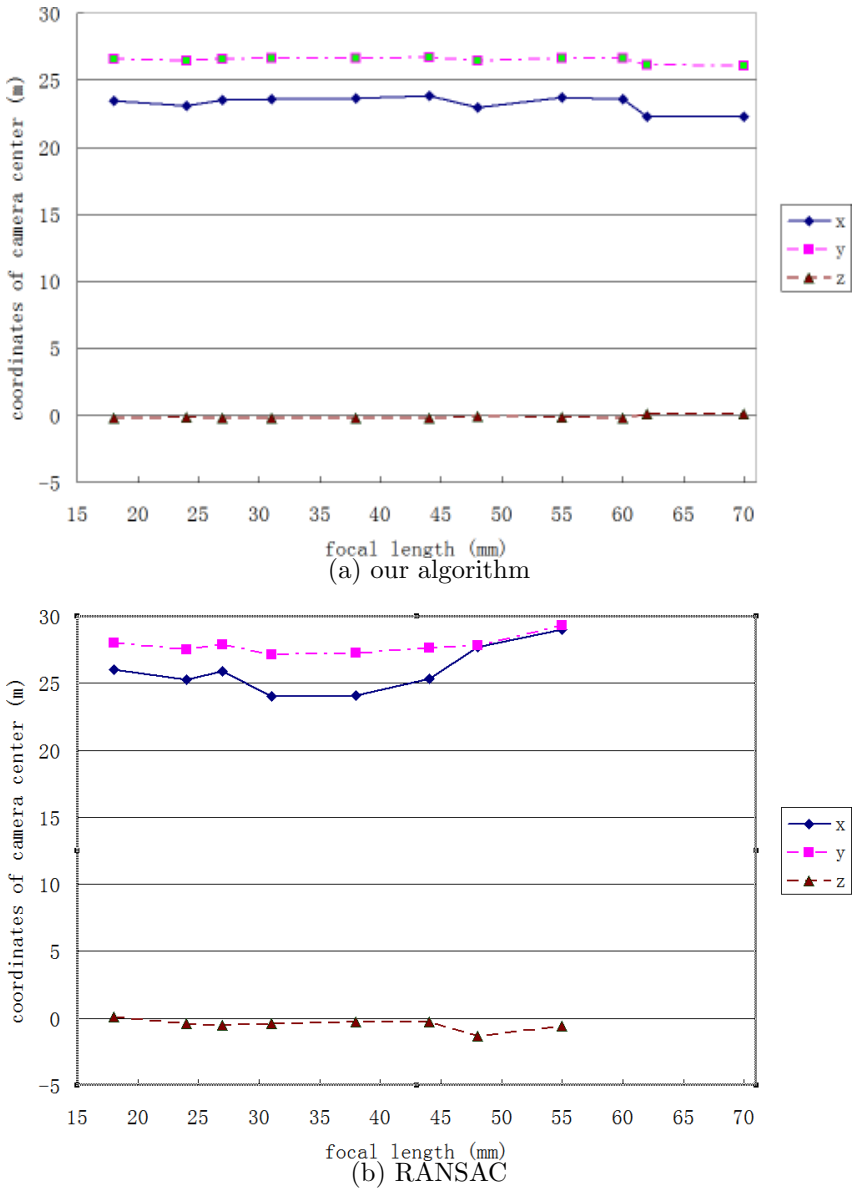


Figure 6.14: Comparison between our algorithm and keypoint matching followed by RANSAC using images at varying focal lengths. (a) shows the x, y, and z coordinate estimates of the camera location obtained by our algorithm across varying focal length, while (b) show the same estimates obtained by the method based on keypoint matching followed by RANSAC. Ideally, the estimate of each coordinate should form a flat horizontal line. Practically, we can see that our algorithm produces consistent and accurate estimates, whereas the other method produces less stable estimates and fails on images with focal length beyond 55mm.

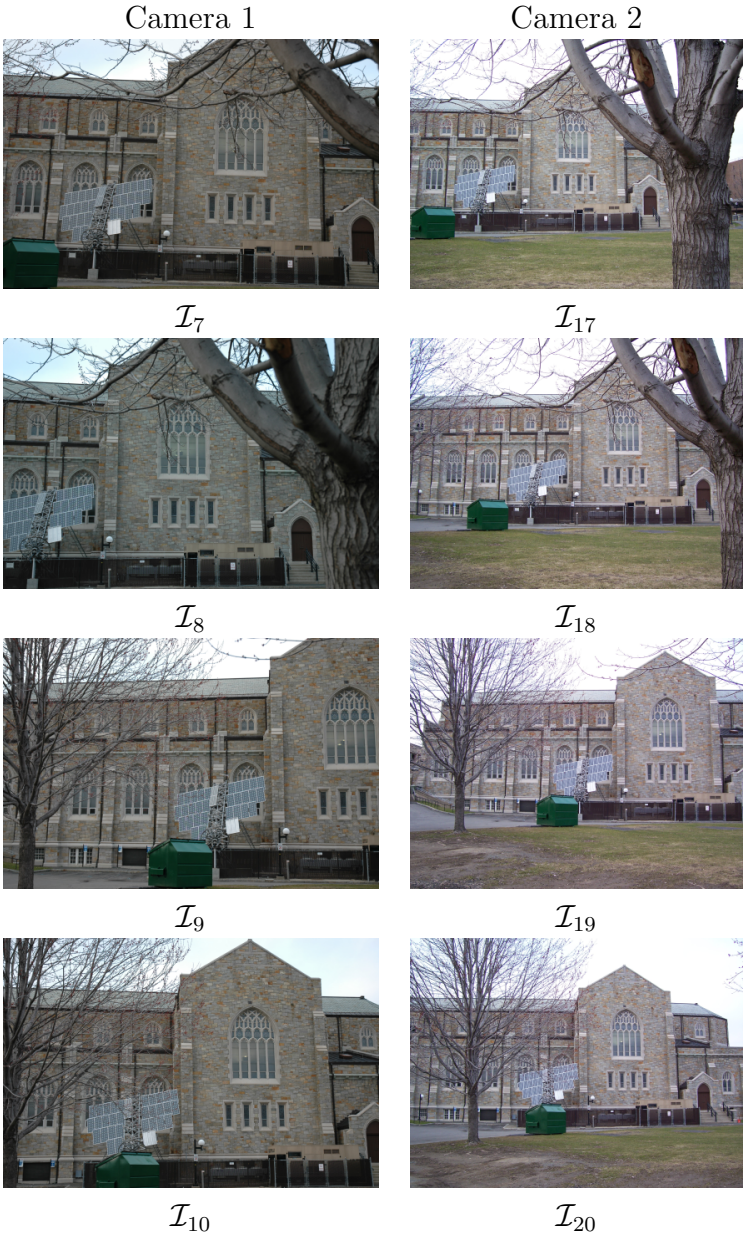
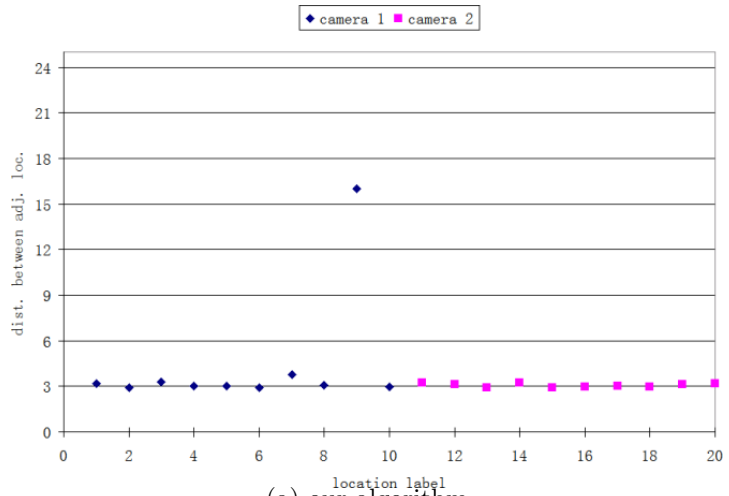
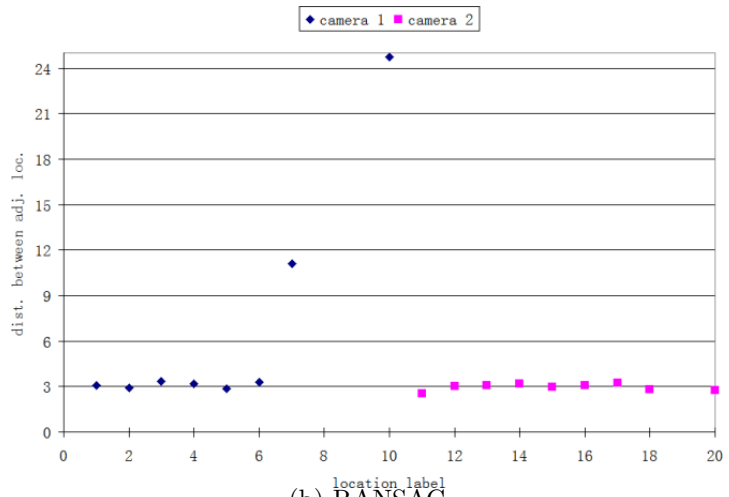


Figure 6.15: Samples of the test images taken from 3-meter-apart locations using two different cameras with fixed zooms. The images in first column were taken by a Kodak CX7530 camera and the ones in the second column were taken by Nikon D70 camera. Both images in each row were taken at the same location and the camera locations are 3-meter-apart between adjacent rows. The corresponding 3d model is the one shown in Figure 6.13d.



(a) our algorithm



(b) RANSAC

Figure 6.16: Comparison between our algorithm and the method based on keypoint matching followed by RANSAC using images taken at 3-meter-apart locations. (a) shows the distance measure between the estimates of adjacent camera locations obtained by our algorithm, while (b) show the same estimates obtained by the method based on keypoint matching followed by RANSAC. The y coordinate represents the distance value between adjacent location estimates, while the x coordinate is the index of images (same as the one used in Figure 6.15). Ideally, these distance measures should form a flat line at $y = 3$. Practically, we can see that the distance measures produced by our algorithm are stable and close to 3m except one image $\mathcal{I}_9$, whereas the other method fails on images between $\mathcal{I}_7$ and $\mathcal{I}_{10}$ and also on image $\mathcal{I}_{19}$.

and percentages, prevents the effective use of RANSAC-style methods.

Next, we analyze briefly how effectively our single-keypoint initialization works. For our 52 successfully-located test images, the refinement algorithm was successful on the first initialization 15 times (29%), within the top five initializations 28 times (54%), and within the top 20 initializations 50 times (96%). From this it is clear that the algorithm can succeed from a small number of correct keypoint matches.

6.4.4 The Effect of Planarity of the Initial Regions

The final consideration in our experiments is to study the effect of planarity of the initial regions on our algorithm. We show this through examples. In Figure 6.11 the visible part of the model is dominated by two planes, while for the image shown in Figure 6.10 with refinement iterations shown in Figure 6.4, even the initial region is non-planar. Interestingly, the algorithm switched to a 3d-to-2d model at about the same iteration during the computation, although the planar region was smaller (the 2nd row of Figure 6.4).

6.4.5 Failures

The eight failures in the 60 test images are mostly due to low contrast and small overlap with the 3d model (shown in Figure 6.17).

In particular, image $\mathcal{I}_9$ shown in Figure 6.15 resulted in an incorrect camera estimate, due to a combination of a close-to-degenerate configuration and "isolated" structures. First, notice the starting plane π is approximately parallel to the image plane (shown in the 1st row of Figure 6.19). As we mentioned before, the camera estimate, when computed using Zhang's planar calibration method, is numerically unstable in this configuration: it is correct only for projecting points on the plane. The projection error becomes large quickly as points move away from the plane. Our algorithm handles this issue by growing off the plane gradually to ensure the projection error remains small enough for correct matching of features. In this particular case, however, when the region grows from the 26th iteration to the 31th iteration, very few constraints are added (this is the isolation area). At the 31st iteration, when matching features in a region (outlined in red in Figure 6.19) that is far from the plane, the error in projection is so large that it results in incorrect

(a)

(b)

Figure 6.17: Two of the failed test images: (a) is low contrast, while (b) has low overlap with the 3d model. The latter will be fixed with the construction of a larger world model.

matches. This leads to the incorrect camera estimate at the end. A checkerboard image of the incorrect camera estimate is shown in Figure 6.18. We can see the dominant plane is well aligned, but not the other regions.

In comparison, image $\mathcal{I}_{19}$ (shown in Figure 6.15), taken from the same location, did not suffer from the same issue. This is because the extra structures to the right provided a smoother way for it to grow off the plane.

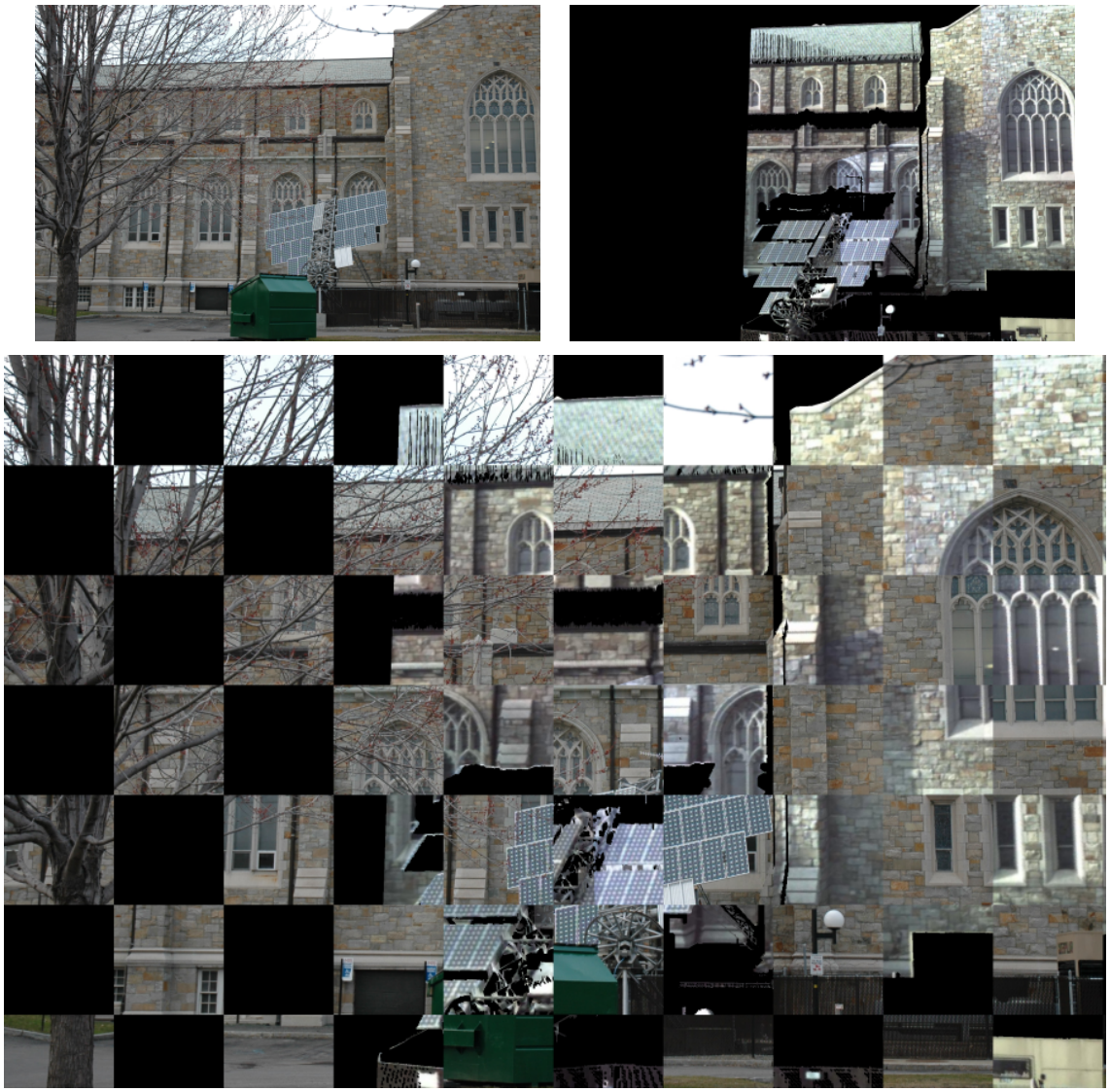


Figure 6.18: A failure example due to isolated structures and a degenerate configuration. The original image is $\mathcal{I}_9$ shown in Figure 6.15.

6.5 Discussion and Conclusions

Our experiments have demonstrated the effectiveness of our approach to locating a test camera image with respect to a 3D model constructed from both LiDAR scans and associated image. Our experimental model was constructed over about a 100m $\times$ 100m area of our campus. No prior information is assumed about camera position and orientation. The algorithm works despite significant differences between model scan acquisition and the test images, including illumination, view-

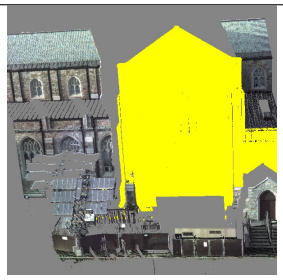
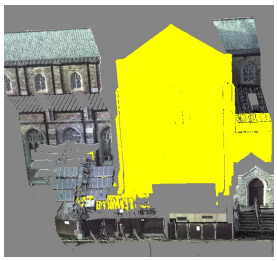
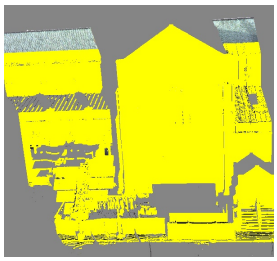
Iter. num.	Model w/ region colored	Test image w/ projected featurs superimposed	Zoom in
26			
30			
31			
36			

Figure 6.19: Several refinement iterations are shown above to demonstrate how the camera estimate get stuck in a local minimal with the presence of isolated structures. The dominant plane on the right is well aligned from the 26th iteration and on. However, starting at the 31th iteration, the features are misaligned in regions with depth a distance away from the plane. The misalignments can be seen from the zoom-ins of two regions outlined in red, which is an indication of the camera estimate stuck at a local minimal.

point and seasonal changes. The few failures involve low image-to-model overlap — sometimes due to occlusions — and substantial illumination changes. Even in these cases, the algorithm correctly indicates that it can not determine the camera location.

The algorithm works within the hypothesize-and-test strategy of the Dual-Bootstrap approach to registration [199]. The primary technical contribution of this chapter is a technique based on model selection for transitioning from a locally-accurate 2d-to-2d model-to-image transformation to a full 3d-to-2d model-to-image projection. A second, more modest contribution is a search for keypoint matches that allows multiple matches to be considered for each test image keypoint and that tests each keypoint match for distinctiveness only locally. This is a first step toward the more general problem of handling models that represent much larger areas. The primary challenges are the increased difficulty of initialization and handling the sheer model size. We have shown, however, that if only one or two good keypoints matches can be found, then our refinement and decision criteria together will likely turn one of them into a correct localization of the camera.

CHAPTER 7
Discussion and Conclusion

7.1 Contributions in Retinal Image Registration

The contributions can be described in two categories: retinal image analysis system (RIVERS) and the generalized image registration.

7.1.1 Contributions in retinal image analysis

In the context of retinal image analysis, the specific contributions are the following:

- **Automatic masking**: We developed a fully-automatic technique for extracting the mask image from a variety of retinal images. The technique combines intensity and edge information, detects the mask boundary from a set of candidates, and refines the boundary used an active contour technique with dynamic programming. The experiments have shown the technique is effective and robust. Quantitatively, compared to the ground-truth mask image, the number of mis-labeled pixels on the generated mask images is within 1% of the total number of pixels. Qualitatively, on 2118 test images, the algorithm only generated 12 incorrect mask images, a failure rate of 0.5%.

- **Covariance-driven refinement**: We developed a refinement technique that is capable of aligning image pairs that the pairwise algorithm fails on during multi-image registration. This technique is complementary to pairwise image registration methods: it uses the initialization from the multi-image alignment registration and searches for new correspondences based on the covariance estimate of the transformation parameters. In the experiments, this technique was applied to several sets of retinal images and was shown successful to align image pairs that had extremely low overlap and did not contain enough information for the pairwise registration to succeed.

7.2 Contributions in Generalized Pairwise Registration

In the second part of the thesis, we presented solutions to two different image registration problems: 1) 2d-to-2d image alignment and 2) camera location estimation.

7.2.1 2d-to-2d Image Alignment

The proposed algorithm, GDB-ICP, uses a hypothesis-and-test strategy and an extension of the Dual-Bootstrap ICP method for refinement. The hypotheses are generated from keypoint matching, which produces a local initial mapping using a similarity transformation. The power to start from a local and low-order transformation reduces the size of the search space for an initial estimate and alleviates the difficulty of obtaining a correct initialization in the presence of various photometric differences. The refinement relies on the Dual-Bootstrap ICP method to "grow" a local initial estimate into a global final estimate using a non-linear transformation. Finally, the novel decision criteria ensure that only the correct transformations are accepted.

GDB-ICP is robust to illumination and scene changes and is capable of aligning images acquired using different modalities, from different view points that produce a small amount of image overlap. When the two images have no overlap or there is insufficient information to determine a stable transformation, the algorithm indicates that the images cannot be aligned instead of producing a false alignment.

We constructed a set of 22 challenging image pairs to test the algorithm, including image pairs with low overlap (e.g. 2%), substantial differences in orientation (90 degrees), and large changes in scale (up to a factor of 6.4). The algorithm successfully registered 19 out of the 22 pairs. When testing with all possible pairs, it rejected 99.8% of the pairs with no overlap.

7.2.2 Camera Location Estimation

As an extension of the algorithmic principle of GDB-ICP, we proposed a 3d-to-2d alignment algorithm to estimate the location of a hand-held camera with respect to a 3d model augmented with texture information. Starting from a local

initial *surface-to-image* mapping, the key issue is how we transition from a 2d-to-2d mapping to a 3d-to-2d projection required for determining the camera location. We addressed this issue by applying a method originally designed for camera calibration using a single planar target. The transition is made using an automatic model selection technique to select the camera model when it is appropriate.

The experiments demonstrated its robustness with respect to illumination changes, physical changes, or occlusion. When applied to a collection of 60 test images and 9 range scans covering 100m x 100m area of RPI campus, our algorithm correctly estimated the camera location of 52 images and indicated that it could not find an camera projection for the remaining 8. On the two sequences of test images, our algorithm clearly surpasseed the methods based on keypoint extraction and matching, followed by RANSAC estimation. On the images these methods fail to locate , our algorithm is able to generate accurate camera estimates.

7.3 Future Work

Though the proposed methods work well on the 2d-to-2d image registration problem and the camera location estimation problem, they do not always succeed and there is still room for improvements.

On the camera location estimation, it is learned that the algorithm presented in Chapter 6 does not work well when starting with an "isolated" surface, especially when the isolated surface is approximately parallel to the image plane. Due to numerical instability, the camera projection estimated from the surface, when applied to other regions of the 3d model, may result in incorrect projected locations. This is analogous to applying extrapolation to points far away from the ones used in the estimation. This issue is solvable by starting with *two* surfaces simultaneously. The constraints on the two surfaces may be combined to produce a more stable camera estimate. However, the combination of two surfaces must be done carefully. Otherwise, it would result in a significant increase of the computation time. For example, 435 hypotheses, as a result of combining every two of the top 30 keypoint matches, is much more expensive to evaluate than the original 30 hypotheses.

Initialization is another area that awaits improvements. The current initializa-

tion scheme, relying on Lowe's keypoint extraction and matching, is not robust to extreme appearance changes, especially on multi-modal images. Kelman, Sofka, and Stewart have made a step towards this direction [101], but the problem is not solved. Importantly, because of the power of the Generalized Dual-Bootstrap algorithm, we need only one correct keypoint match to succeed. The fact that no combinations of keypoint matches are needed is very different from the commonly-used methods based on keypoint matching, followed by the application of RANSAC. Hence, the question becomes how we can obtain *one* correct keypoint match *efficiently*. This is particularly important for the camera location estimation problem, because after demonstrating the capabilities of our location estimation algorithm, the key issue becomes efficiently matching image keypoints when there is a huge number of keypoints in an integrated model.

Another important piece of the work is to broaden the application domain of the unified approach for pairwise image registration. In particular, parallax and deformable image registration are two important problems to be solved.

Parallax, or fundamental matrix estimation problem, is an important step towards structure from motion. There have been significant improvements on the wide-base line matching algorithms during the last decade [24, 45, 133, 143, 170, 225]. However, these methods are not particularly robust to appearance changes or physical changes. One idea is to apply the unified approach using the piecewise-homography models. Each provides constraints on the fundamental matrix. Because this method relies on having planar regions present in the image, however, the overall effectiveness of this method is to be evaluated.

Deformable image registration, on the other hand, posts different challenges. In particular, the large number of parameters in the deformable transformation could result in inaccurate computation of the covariance matrix, and hence impedes the effective use of the Dual-Bootstrap refinement method. This question is to be answered in the future work.

7.4 Conclusion

This thesis has described four novel methods in the context of the image registration.

Most importantly, we proposed two algorithms for solving the 2d-to-2d image alignment problem and the camera location estimation problem. These algorithms have proven robust to appearance changes, physical changes, viewpoint changes, occlusions, and small overlaps. The 2d-to-2d image registration algorithm has also been shown effective with several different image modalities.

In addition, in the context of retinal image registration, an automatic masking technique was developed to extract retinal mask images. This technique is currently in use with retinal image feature extraction, image registration, and change detection algorithms. A covariance-driven refinement algorithm was developed to handle images with low overlaps in the context of multi-image registration.

Bibliography

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. *ACM Transactions on Graphics*, 23(3):292–300, 2004. 20, 45, 46

[2] M. Ahmed and A. Farag. Non-metric calibration of camera lens distortion. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 157–160, Oct. 2001. 43

[3] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:855–867, 1990. 47, 56, 57, 58

[4] Y. Amit. A nonlinear variational problem for image matching. *SIAM Journal on Scientific Computing*, 15(1):207–224, 1994. 40

[5] X. Armangue, J. Salvi, and J. Batlle. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, 35(7), 2002. 40, 43

[6] N. Asada, A. Amano, and M. Baba. Photometric calibration of zoom lens systems. In *International conference on pattern recognition (ICPR 96)*, volume 1, pages 186–190, 1996. 37

[7] S. Aylward, J. Jomier, S. Weeks, and E. Bullitt. Registration of vascular images. *International Journal of Computer Vision*, 55(2-3):123–138, 2003. 25

[8] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004. 30, 31

[9] D. I. Barnea and H. F. Silverman. A class of algorithms for fast digital image registration. *IEEE Transactions on Computers*, C-21(2):179–186, February 1972. 25, 30

[10] A. E. Beaton and J. W. Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16:147–185, 1974. 51, 78

[11] D. E. Becker, A. Can, H. L. Tanenbaum, J. N. Turner, and B. Roysam. Image processing algorithms for retinal montage synthesis, mapping, and real-time location determination. *IEEE Transactions on Biomedical Engineering*, 45(1):105–118, 1998. 9

[12] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color- and texture-based image segmentation using EM and its application to content-based image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 675–682, 1998. 86

[13] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, June 2002. 29, 123

[14] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the Second European Conference on Computer Vision*, pages 237–252, 1992. 31, 32

[15] R. Bergevin, D. Laurendeau, and D. Poussart. Registering range views of multipart objects. *Computer Vision and Image Understanding*, 61(1):1–16, 1995. 35, 38

[16] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multiview registration technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):540–547, 1996. 27

[17] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 25, 27, 28

[18] M. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow-fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996. 31

[19] G. Blais and M. Levine. Registering multiview range data to create 3d computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):820–824, August 1995. 27

[20] R. C. Bolles and M. A. Fischler. A Ransac-based approach to model fitting and its application to finding cylinders in range data. In *Proceedings Seventh International Joint Conference on Artificial Intelligence*, pages 637–643, 1981. 29

[21] K. Bowyer, C. Kranenburg, and S. Dougherty. Edge detector evaluation using empirical roc curves. *Computer Vision and Image Understanding*, 84:77–103, 2001. 47

[22] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, December 1992. 25, 35, 37

[23] M. Brown and D. Lowe. Recognising panoramas. In *Proceedings of the IEEE International Conference on Computer Vision*, 2003. 1, 20, 29, 30, 44, 45, 46, 68, 108, 123

[24] M. Brown and D. G. Lowe. Unsupervised 3d object recognition and reconstruction in unordered datasets. In *Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*, 2005. 166

[25] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 510–517, 2005. 29, 105, 127

[26] K. Bubna and C. V. Stewart. Model selection techniques and merging rules for range data segmentation algorithms. *Computer Vision and Image Understanding*, 80:215–245, 2000. 100

[27] K. P. Burnham and D. R. Anderson. *Model Selection and Inference: A practical Information-theorectic Approach*. Springer, 1st edition, 1998. 100, 101, 135

[28] P. J. Burt and E. H. Adelson. A multiresolution spline with applications to image mosaics. *tgraph*, 2(217-236):4, 83. 20, 46

[29] A. Can, H. Shen, J. N. Turner, H. L. Tanenbaum, and B. Roysam. Rapid automated tracing and feature extraction from live high-resolution retinal fundus images using direct exploratory algorithms. *IEEE Transactions on Information Technology in Biomedicine*, 3(2):125–138, 1999. 4, 7, 9

[30] A. Can, C. Stewart, and B. Roysam. Robust hierarchical algorithm for constructing a mosaic from images of the curved human retina. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 286–292, 1999. 7

[31] A. Can, C. Stewart, B. Roysam, and H. Tanenbaum. A feature-based algorithm for joint, linear estimation of high-order image-to-mosaic transformations: Mosaicing the curved human retina. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):412–419, 2002. 3, 6, 7, 9, 67, 68, 73, 74

[32] A. Can, C. Stewart, B. Roysam, and H. Tanenbaum. A feature-based, robust, hierarchical algorithm for registering pairs of images of the curved human retina. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):347–364, 2002. 4, 7, 9, 25, 27, 28, 38, 70, 72, 79, 118

[33] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, Nov. 1986. 47, 52

[34] D. Capel and A. Zisserman. Automated mosaicing with super-resolution zoom. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 885–891, 1998. 1, 44, 45

[35] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–139, 1990. 44

[36] G. Champleboux, S. Lavallee, R. Szeliski, and L. Brunie. From accurate range imaging sensor calibration to accurate model-based 3-d object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 83–89, 1992. 25, 27

[37] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, Feb. 2001. 55

[38] C. Chatterjee and V. P. roychowdhury. Algorithms for coplanar camera calibration. *Machine Vision Applications*, 12:84–97, 2000. 43, 138

[39] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992. 25, 27, 28

[40] H. Chui and A. Rangarajan. A feature registration framework using mixture models. In *Proceedings of Mathematical Methods in Biomedical Image Analysis*, pages 190–197, 2000. 28

[41] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2–3):114–141, 2003. 28

[42] H. Chui, A. Rangarajan, J. Zhang, and C. M. Leonard. Unsupervised learning of an atlas from unlabeled point-sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):160–172, 2004. 28, 122

[43] K. Cios, editor. *Medical Data Mining and Knowledge Discovery*, chapter ADRIS: an Automatic Diabetic Retinal Image Screening System, pages 181–210. Springer Verlag, 2001. 3

[44] M. M. Coselmon, J. M. Balter, D. L. McShan, and M. L. Kessler. Mutual information based ct registration of the lung at exhale and inhale breathing states using thin-plate splines. *Medical Physics*, 31(11):2942–2948, 2004. 35

[45] G. Cross, A. W. Fitzgibbon, and A. Zisserman. Parallax geometry of smooth surfaces in multiple views. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pages 323–329, 1999. 166

[46] C. M. Cyr, T. B. Sebastian, and B. B. Kimia. 2D-3D registration based on shape matching. In *Proceedings of Mathematical Methods in Biomedical Image Analysis*, 2000. 44

[47] J. Davis. Mosaics of scenes with moving objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 354–360, 1998. 45

[48] P. E. Debevec and J. malik. Recovering high dynamic range radiance maps from photographs. In *Computer Graphics Proceedings (SIGGRAPH)*, Los Angeles, California, USA, Aug. 1997. 1

[49] M. DeGrezia and M. Robinson. Ophthalmic manifestations of HIV: an update. *Journal of the Association of Nurses in AIDS Care*, 12(3):22–32, May-Jun 2001. 13, 68

[50] F. Devernay and O. Faugeras. Straight lines have to be straight. *Machine Vision Applications*, 13(1):14–24, 2001. 43

[51] Diabetic Retinopathy Research Group. Diabetic retinopathy study. Report Number 7. A modification of the Airlie House classification of diabetic retinopathy. *Investigative Ophthalmology & Visual Science*, 21:210–226, Dec 1981. 4, 12

[52] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, 2001. 28

[53] Y. Dufournaud, C. Schmid, and R. Horaud. Image matching with scale adjustment. *Computer Vision and Image Understanding*, 93:175–194, 2004. 29, 120

[54] Early Treatment Diabetic Retinopathy Study Research Group. Results from the early treatment diabetic retinopathy study. *Ophthalmology*, 98:739–840, 1991. 3

[55] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, 2nd edition, 1990. 61

[56] O. Faugeras, Q.-T. Luong, and S. Maybank. Camera self-calibration: Theory and experiments. In *Proceedings of the Second European Conference on Computer Vision*, 1992. 43

[57] J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. *International Journal of Computer Vision*, 18(2):99–119, 1996. 40

[58] V. Ferrari, T. Tuytelaars, and L. V. Gool. Simultaneous object recognition and segmentation by image exploration. In *Proceedings of the Eigth European Conference on Computer Vision*, 2004. 122

[59] R. Fransens, C. Strecha, and L. V. Gool. Multimodal and multiband image registration using mutual information. In *ESA-EUSC*, 2004. 85

[60] F. Fraundorfer and H. Bischof. Global localization from a single feature correspondence. In *Digital Imaging and Pattern Recognition, 30th Workshop of the Austrian Association for Pattern Recognition*, pages 151–160, 2006. 30, 40, 42, 125

[61] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, September 1991. 29

[62] R. W. Frischholz and K. P. Spinnler. Class of algorithms for real-time subpixel registration. In D. W. Braggins, editor, *Proceedings of SPIE Conference on Computer Vision for Industry*, pages 50–59, Dec. 1993. 31

[63] K. H. Fritzsche. *Computer Vision Algorithms for Retinal Vessel Detection and Width Change Detection*. PhD thesis, Rensselaer Polytechnic Institute, Troy, New York, Dec 2004. 7, 9

[64] H. Gagnon, M. Soucy, R. Bergevin, and D. Laurendeau. Registration of multiple range views for automatic 3-d model building. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 581–586, 1994. 27

[65] L. Gagnon, M. Lalonde, M. Beaulieu, and M.-C. Boucher. Procedure to detect anatomical structures in optical fundus images. In *Proceedings of SPIE Conference Medical Imaging: Image Processing*, volume 4322, pages 1218–1225, 2001. 9

[66] A. Georgiev and P. K. Allen. Visoin for mobile robot localization in urban environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 472–477, Washington, USA, May 2002. 40, 42

[67] C. A. Glasbey and K. V. Mardia. A review of image-warping methods. *Journal of Applied Statistics*, 25(2):155–, apr 1998. 35

[68] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, 3rd edition edition, 1992. 32

[69] L. V. Gool, T. Moons, and D. Ungureanu. Affine/ photometric invariants for planar intensity patterns. In *Proceedings of the Fourth European Conference on Computer Vision*, 1996. 29

[70] A. Goshtasby. Piecewise linear mapping functions for image registration. *Pattern Recognition*, 19:459–466, 1986. 40

[71] A. Goshtasby. Piecewise cubic mapping functions for image registration. *Pattern Recognition*, 20:525–533, 1987. 40

[72] R. Graham. An efficient algorithm for determining the convex hull of a finite planar point set. In *Info. Proc. Letters*, volume 1, pages 132–133, 1972. 53

[73] S. Granger and X. Pennec. Multi-scale EM-ICP: A fast and robust approach for surface registration. In *Proceedings of the Seventh European Conference on Computer Vision*, pages 418–432, 2002. 28, 122

[74] P. J. Green. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(2):149–192, 1984. 50

[75] W. Grimson, T. Lozano-Perez, W. Wells, G. Ettinger, and S. White. An automatic registration method for frameless stereotaxy, image, guided surgery and enhanced reality visualization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 430–436, 1994. 27

[76] F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley and Sons, 2001. 1

[77] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, page 979984, 1999. 40, 42

[78] X. Han, C. Xu, and J. L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):755–768, 2003. 55

[79] M. J. Hannah. A system for digital stereo image matching. *Photogrammetric Engineering and Remote Sensing*, 55(12):1765–1770, 1989. 32

[80] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, Manchester, UK, 1988. 27, 86, 93

[81] J. A. Hartigan. *Clustering Algorithms*. Wiley, NY, 1975. 47

[82] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000. 37, 39, 77, 100

[83] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer. Comparison of edge detectors: a methodolody and initial study. *Computer Vision and Image Processing*, 69(1):38–54, Jan. 1998. 47

[84] T. Hellstedt, E. Vesti, and I. Immonen. Identification of individual microaneurysms: a comparison between flourescein angiograms and red-free colour photographs. *Graefes Archive for Clinical and Experimental Ophthalmology*, 234:13–17, 1996. 4

[85] E. E. Hemayed. A survey of camera self-calibration. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 351–357, Miami, Florida, July 2003. 43

[86] S. Henn and K. Witsch. Multimodal image registration using a variational approach. *SIAM Journal on Scientific Computing*, 25:1429–1447, 2003. 40

[87] G. Hermosillo, C. Chefd'Hotel, and O. Faugeras. Variational methods for multimodal image matching. *ijcv*, 50(3):329–343, 2002. 40

[88] P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Commun. Statist.-Theor. Meth.*, A6:813–827, 1977. 98

[89] A. Hoover and M. Goldbaum. Locating the optic nerve in a retinal image using the fuzzy convergence of the blood vessels. *IEEE Transactions on Medical Imaging*, 22(8):951–958, 2003. http://www.parl.clemson.edu/stare/nerve/. 9, 11

[90] C.-T. Hsu, T.-H. Cheng, R. A. Beuker, and J.-K. Horng. Feature-based video mosaic. In *Proceedings of the IEEE International Conference on Image Processing*, pages 887–890, 2000. 68

[91] T. S. Huang and A. N. Netravali. Motion and structure from feature correspondences: A review. *Proceedings of the IEEE*, 82(2):252–267, Feb 1994. 68

[92] D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment. *International Journal of Computer Vision*, 5(2):195–212, 1990. 126

[93] M. Irani and P. Anandan. Robust multisensor image alignment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 959–966, 1998. 32

[94] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 605–611, 1995. 1, 45, 68

[95] Y. Iwashita, R. Kurazume, T. Hasegawa, K. konishi, M. Nakamoto, and M. Hashizume. Fast 2D-3D registration for navigation system of surgical robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Apr. 2005. 21, 40, 44

[96] P. J.P.W., M. J.B.A., and V. M.A. Mutual information matching in multiresolution contexts. *Image and Vision Computing*, 19(1):45–52, 2001. 35

[97] S. Ju, M. Black, and A. Jepson. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 307–314, 18–20 June 1996. 40

[98] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *Proceedings of the Eigth European Conference on Computer Vision*, 2004. 29

[99] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier, 1996. 100

[100] M. Kass, A. P. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, Jan. 1988. 47, 55

[101] A. Kelman, M. Sofka, and C. Stewart. Keypoint descriptors for matching across multiple image modalities and non-linear intensity variations. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Workshop on Image Registration and Fusion*, Minneapolis, MN, 23June 2007. 166

[102] R. Kimmel and A. M. Bruckstein. Tracking level sets by level sets: A method for solving the shape from shading problem. *Computer Vision and Image Understanding*, 62(2):47–58, July 1995. 55

[103] B. J. King, T. Malisiewicz, C. V. Stewart, and R. J. Radke. Registration of multiple range scans as a location recognition problem: Hypothesis generation, refinement and verification. In *Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*, pages 180–187, 2005. 38, 150

[104] E. Kohner, I. Sleightholm, S. Aldington, R. Turner, and D. Matthews. Microaneurysms in the development of diabetic retinopathy. *Diabetologia*, 42:1107–1112, 1999. 4

[105] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu. Statistical edge detection: Learning and evaluating edge cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):57–74, Jan. 2003. 47

[106] H. M. Kooy, M. van Herk, P. D. Barnes, and E. Alexander. Image fusion for stereotactic radiotherapy and radiosurgery treatment planning. *Int. J. Radiat. Oncol. Biol. Phys.*, 28(5):1229–1234, 1994. 1

[107] J. M. Krauss and C. A. Puliafito. Lasers in ophthalmology. *Lasers In Surgery And Medicine*, 17:102–159, 1995. 4

[108] R. Kumar, P. Anandan, M. Irani, J. Bergen, and K. Hanna. Representation of scenes from collections of images. In *Representation of Visual Scenes, 1995. (In Conjuction with ICCV'95), Proceedings IEEE Workshop on*, pages 10–17, 1995. 38

[109] R. Kumar, H. S. Sawhney, J. C. Asmuth, A. Pope, and S. Hsu. Registration of video to geo-referenced imagery. In *International conference on pattern recognition (ICPR 98)*, volume 2, pages 1393–4100, 1998. 68

[110] J. Kybic. High-dimensional mutual information estimation for image registration. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 1779–1782, Singapore, Oct.24–27 2004. 34

[111] K. K. Lau and A. C. S. Chung. A global optimization strategy for 3d-2d registration of vascular images. In *Proceedings of the British Machine Vision Conference*, 2006. 40, 44

[112] Leica Geosystems. The HDS product family. http://www.leica-geosystems.com/hds/en/lgs_62189.htm. Last accessed 1 June 2007. 124

[113] K. Levenberg. A method for the solution of certain problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944. 135, 138

[114] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):418–433, 2005. 68

[115] B. Likar and F. Pernus. A hierarchical approach to elastic registration based on mutual information. *Image and Vision Computing*, 19(1):33–44, 2001. 35

[116] G. Lin, C. V. Stewart, B. Roysam, K. Fritzsche, and G. Yang. Predictive scheduling algorithms for real-time feature extraction and spatial referencing: Application to retinal image sequences. *IEEE Transactions on Biomedical Engineering*, 51:115–124, 2004. 4

[117] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994. 117

[118] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30:79–116, Nov. 1998. 93

[119] L. Liu and I. Stamos. Automatic 3D to 2D registration for the photorealistic rendering of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 41

[120] L. Liu, I. Stamos, G. Yu, G. Wolberg, and S. Zokai. Multiview geometry for texture mapping 2d images onto 3d range data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, 2006. 41, 124, 125

[121] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987. 126

[122] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004. 29, 30, 42, 86, 88, 127, 128, 130, 151, 153

[123] C.-P. Lu, G. D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, 2000. 40, 41

[124] B. Lucas and T. Kanade. An iteractive image registration technique with an application to stereo vision. In *Proceedings Seventh International Joint Conference on Artificial Intelligence*, pages 674–679, 1981. 30

[125] B. Luo and E. R. Hancock. Iterative procrustes alignment with the EM algorithm. *Image and Vision Computing*, 20(5-6):377–396, Apr. 2002. 28

[126] Q.-T. Luong and O. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17(1):43–75, 1996. 38

[127] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16(2):187–198, 1997. 32, 34, 35

[128] F. Maes, D. Vandermeulen, and P. Suetens. Comparative evaluation of multiresolution optimization strategies for multimodality image registration by maximization of mutual information. *Medical Image Analysis*, 3(4):373–386, 1999. 35

[129] J. A. Maintz, P. A. van den Elsen, and M. A. Viergever. Comparison of edge-based and ridge-based registration of CT and MR brain images. *Medical Image Analysis*, 1(2):151–161, 1996. 25

[130] E. Malis and R. Cipolla. Multi-view constraints between collineations: Application to self-calibration from unknown planar structures. In *Proceedings of the Sixth European Conference on Computer Vision*, 2000. 43

[131] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, Feb. 1995. 55

[132] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, pages 431–441, 1963. 100, 135, 138

[133] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, 2002. 166

[134] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, Sept. 2004. 29, 30

[135] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996. 39

[136] R. A. McLaughlin, J. Hipwell, D. J. Hawkes, J. A. Noble, J. V. Byrne, and T. C. Cox. A comparison of a similarity-based and a feature-based 2-dc3-d registration method for neurointerventional use. *IEEE Transactions on Medical Imaging*, 24(8):1058–1066, 2005. 44

[137] P. Meer. Robust techniques for computer vision. In G. Medioni and S. B. Kang, editors, *Emerging Topics in Computer Vision*. Prentice Hall, 2004. 50, 98

[138] P. Meer, D. Mintz, and A. Rosenfield. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6(1):59–70, 1991. 29

[139] Y. Meng and H. Zhuang. What you see is what you get — self-calibrating camera lens distortion. *IEEE Journal of Robotics and Automation*, 11(4):123–127, 2004. 37

[140] C.-H. Menq, H.-T. Yau, and G.-Y. Lai. Automated precision measurement of surface profile in CAD-directed inspection. *IEEE Transactions on Robotics and Automation*, 8(2):268–278, 1992. 28

[141] C. R. Meyer, J. L. Boes, B. Kim, P. H. Bland, and K. R. Zasadny. Demonstration of accuracy and clinical versatility of mutual information for automatic multimodality image fusion using affine and thin-plate spline warped geometric deformations. *Medical Image Analysis*, 1(3):195–206, 1996/7. 1

[142] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages II:257–263, 2003. 88

[143] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004. 2, 29, 86, 88, 117, 166

[144] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. 29

[145] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1–2):43–72, 2005. 29

[146] D. Milgram. Computer methods for creating photomosaics. *IEEE Transactions on Computers*, C-24(11):1113–1119, November 1975. 25

[147] J. V. Miller. *Regression-Base Surface Reconstruction: Coping with Noise, Outliers, and Discontinuities*. PhD thesis, Rensselaer Polytechnic Institute, Aug. 1997. 29, 99, 108

[148] J. V. Miller and C. V. Stewart. MUSE: Robust surface fitting using unbiased scale estimates. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 300–306, 18–20 June 1996. 78

[149] P. N. Monahan, K. A. Gitter, J. D. Eichler, and G. Cohen. Evaluation of persistence of subretinal neovascular membranes using digitized angiographic analysis. *Retina—The Journal of Retinal and Vitreous Diseases*, 13(3):196–201, 1993. 4

[150] P. N. Monahan, K. A. Gitter, J. D. Eichler, G. Cohen, and K. Schomaker. Use of digitized fluorescein angiogram system to evaluate laser treatment for subretinal neovascularization: Technique. *Retina—The Journal of Retinal and Vitreous Diseases*, 13(3):187–195, 1993. 4

[151] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995. 42

[152] R. Murphy. Age-related macular degeneration. *Ophthalmology*, 9:696–971, 1986. 4

[153] H. Narasimha-Iyer, A. Can, B. Roysam, C. V. Stewart, H. L. Tanenbaum, A. Majerovics, and H. Singh. Robust classification of longitudinal changes in color retinal fundus images for monitoring diabetic retinopathy,. *IEEE Transactions on Biomedical Engineering*, 53(6), 2006. 9, 66

[154] J. Neter and W. Wasserman. *Applied Linear Statistical Models*. Richard D. Irwin Inc., 3rd edition, 1975. 74

[155] L. P. Nolte, M. A. Slomczykowski, U. Berlemann, M. J. Strauss, R. Hofstetter, D. Schlenzka, T. Laine, and T. Lund. A new approach to computer-aided spine surgery: fluoroscopy-based surgical navigation. *European Spine journal*, 9(7), Feb. 2000. 21

[156] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*. Springer-Verlag, 2003. 55

[157] S. Osher and R. P. Fedkiw. Level set methods: An overview and some recent results. Lecture in workshops of the GBM program, September 2001. 55

[158] D. C. P.Ñewman and K. Ho. Outdoor slam using visual appearance and laser ranging. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006. 40, 41, 42, 125

[159] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, 2000. 55

[160] S. Peleg and J. Herman. Panoramic mosaics by manifold projection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 338–343, 1997. 13, 68

[161] S. Periaswamy and H. Farid. Medical image registration with partial data. *Medical Image Analysis*, In Press, Corrected Proof, 2005. 40

[162] G. Piella. A general framework for multiresolution image fusion: from pixels to regions. *Information Fusion*, 4(4):259–280, 2003. 1

[163] A. Pitiot, E. Bardinet, P. M. Thompson, and G. Malandain. Piecewise affine registration of biological images for volume reconstruction. *Medical Image Analysis*, 10(3):465–483, 2005. 40

[164] J. P. W. Pluim, J. B. A. Maintz, and M. A. Vierveger. Image registration by maximization of combined mutual information and gradient information. *IEEE Transactions on Medical Imaging*, 19(8):809–814, Aug 2000. 34, 85

[165] J. P. W. Pluim, J. B. A. Maintz, and M. A. Vierveger. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004, 2003. 32, 33, 34

[166] C. Pohl and J. L. Genderen. Multisensor image funsion in remote sensing: concepts, methods and applications. *International Journal of Remote Sensing*, 19(5):823–854, 1998. 1

[167] M. Pollefeys, R. Koch, and L. J. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999. 43

[168] H. Pottmann, S. Leopoldseder, and M. Hofer. Registration without icp. *Computer Vision and Image Understanding*, 95(1):54–71, July 2004. 28

[169] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992. 74

[170] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 754–760, 1998. 2, 166

[171] G. Qian and R. Chellappa. Structure from motion using sequential monte carlo methods. *International Journal of Computer Vision*, 59(1), 2004. 68

[172] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Transactions on Image Processing*, 14(3):294–307, 2005. 1, 3, 9, 20, 44, 66

[173] D. Robertson and R. Cipolla. An image-based system for urban navigation. In *Proceedings of the British Machine Vision Conference*, 2004. 40, 41, 42, 124, 125

[174] A. Roche, X. Pennec, G. Malandain, and N. Ayache. Rigid registration of 3-d ultrasound with MR images: a new approach combining intensity and gradient information. *IEEE Transactions on Medical Imaging*, 20(10):1038–1049, 2001. 35

[175] T. Rohlfing, C. Maurer, Jr., D. Bluemke, and M. Jacobs. Volume-preserving non-rigid registration of mr breast images using free-form deformation with an incompressibility constraint. *IEEE Transactions on Medical Imaging*, 22(6), 2003. 40

[176] D. Rueckert. Nonrigid registration: concepts, algorithms, and applications. In J. V. Hajnal, D. L. G. Hill, and D. J. Hawkes, editors, *Medical Image Registration*, pages 281–302. CRC Press, 2001. 39

[177] D. Rueckert, I. Somoda, C. Hayes, D. Hill, M. Leach, and D. Hawkes. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Transactions on Medical Imaging*, 18:712–721, 1999. 40

[178] S. Sakuma, T. Nakanishi, and Y. Takahashi. Automated detection of changes in sequential color ocular fundus images. In *Proceedings of SPIE Conference on Image Processing*, volume 3338, pages 1377–1385, 1998. 3

[179] H. Sawhney, S. Hsu, and R. Kumar. Robust video mosaicing through topology inference and local to global alignment. In *Proceedings of the Fifth European Conference on Computer Vision*, volume II, pages 103–119, 1998. 1, 13, 31, 68, 85

[180] H. Sawhney and R. Kumar. True multi-image alignment and its application to mosaicing and lens distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):235–243, 1999. 1, 13, 36, 37, 44, 45, 68

[181] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or How do I organize my holiday snaps? In *Proceedings of the Seventh European Conference on Computer Vision*, volume 1, pages 414–431, 2002. 13, 68, 125

[182] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000. 86

[183] S. Se, D. G. Lowe, and J. J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005. 30, 40, 42, 125

[184] J. K. Seo, G. C. Sharp, and S. W. Lee. Range data registration using photometric features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 27

[185] J. A. Sethian. *Level set methods and fast marching methods*. Cambridge university press, 2 edition, 1999. 55

[186] Y. Shan, H. S. Sawhney, and R. Kumar. Unsupervised learning of discriminative edge measures for vehicle matching between non-overlapping cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 123

[187] E. Shechtman, Y. Caspi, and M. Irani. Space-time super resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):531–545, april 2005. 1, 44

[188] R. Shekhar and V. Zagrodsky. Mutual information-based rigid and nonrigid registration of ultrasound volumes. *IEEE Transactions on Medical Imaging*, 21(1):9–22, Jan. 2002. 35

[189] D. Shen and C. Davatzikos. HAMMER: Hierarchical Attribute Matching Mechanism for Elastic Registration. *IEEE Transactions on Medical Imaging*, 21(11):1421–1439, 2002. 40, 94

[190] H. Shen, C. Stewart, B. Roysam, G. Lin, and H. Tanenbaum. Frame-rate spatial referencing based on invariant indexing and alignment with application to laser retinal surgery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):379–384, March 2003. 4

[191] H. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130, 2000. 1, 13, 32, 44, 45, 68, 69, 70

[192] H.-Y. Shum and R. Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 953–958, 1998. 45

[193] M. Sofka and C. V. Stewart. Retinal vessel extraction using multiscale matched filters, confidence and edge measures. *IEEE Transactions on Medical Imaging*, 25(12):1531–1546, Dec. 2006. 13

[194] I. Stamos and P. K. Allen. Automatic registration of 2-d with 3-d imagery in urban environments. In *Proceedings of the IEEE International Conference on Computer Vision*, 2001. 41, 124, 125

[195] I. Stamos and P. K. Allen. Geometry and texture recovery of scenes of large scale. *Computer Vision and Image Understanding*, 88:94–118, 2002. 21

[196] D. Steedly, C. Pal, and R. Szeliski. Efficiently registering video into panoramic mosaics. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 1300–1307, 2005. 45

[197] K. L. Steele and P. K. Egbert. Correspondence expansion for wide baseline stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 122

[198] G. Stein. Lens distortion calibration using point correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 602–608, 1997. 37

[199] C. Stewart, C.-L. Tsai, and B. Roysam. The Dual-Bootstrap Iterative Closest Point algorithm with application to retinal image registration. *IEEE Transactions on Medical Imaging*, 22(11):1379–1394, 2003. 6, 7, 9, 21, 66, 67, 70, 72, 79, 87, 100, 101, 105, 118, 122, 126, 162

[200] C. V. Stewart. Robust parameter estimation in computer vision. *SIAM Reviews*, 41(3):513–537, 1999. 50, 72, 93, 98, 134

[201] A. Stoddart, S. Lemke, A. Hilton, and T. Renn. Estimating pose uncertainty for surface registration. *Image and Vision Computing*, 16(2):111–120, 1998. 28

[202] C. Studholme, D. Hill, and D. Hawkes. An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognition*, 32:71–86, 1999. 34

[203] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proceedings of the Fourth European Conference on Computer Vision*, volume 2, pages 709–720, 1996. 68

[204] P. F. Sturm and S. J. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 23–25, 1999. 40, 43, 136, 138

[205] M.-S. Su, W.-L. Hwang, and K.-Y. Cheng. Analysis on multiresolution mosaic images. *IEEE Transactions on Image Processing*, 2004. 20, 46

[206] R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, 16(2):22–30, 1996. 1

[207] R. Szeliski. Image alignment and stitching. Technical report, Microsoft Research, 2004. 7, 20, 45

[208] R. Szeliski and S. Kang. Recovering 3D shape and motion from image streams using nonlinear least squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28, 1994. 45

[209] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and texture-mapped models. In *Computer Graphics Proceedings (SIGGRAPH)*, pages 251–258, Los Angeles, California, USA, August 1997. 45

[210] T. Tamaki, T. Yamamura, and N. Ohnishi. Unified approach to image distortion. In *International conference on pattern recognition (ICPR'02)*, volume 2, pages 584– 587, 2002. 37

[211] T. Teng, M. Lefley, and D. Claremont. Progress towards automated diabetic ocular screening: a review of image analysis and intelligent systems for diabetic retinopathy. *IEE Medical & Biological Engineering & Computing*, 40(1):2–13, Jan 2002. 3

[212] D. Terzopoulos and D. Metaxas. Dynamic 3d models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, 1991. 39

[213] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123, 1988. 39

[214] P. Thevenaz and M. Unser. Optimization of mutual information for multiresolution image registration. *IEEE Transactions on Image Processing*, 9(12):2083–2099, Dec 2000. 40

[215] Q. Tian and M. N. Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics, and Image Processing*, 35:220–233, 1986. 31

[216] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, Carnegie Mellon Univ., 1991. 32

[217] P. Torr. An assessment of information criteria for motion model selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 47–52, 1997. 100

[218] P. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, April 2000. 108

[219] J. Townshend, C. Justice, C. Gurney, and J. McManus. The impact of misregistration on change detection. In *Geoscience and Remote Sensing, IEEE Transactions on*, volume 30, pages 1054–1060, 1992. 1

[220] B. Triggs, P. MacLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment — a modern synthesis. In *Vision Algorithms: Theory and Practice, LNCS 1883*, pages 298–372. Springer, 1999. 45, 68, 69

[221] A. Troccoli and P. K. Allen. A shadow based method for image to model registration. In *IEEE Workshop on Image and Video Registration*, Washington DC, USA, July 2004. 125

[222] C.-L. Tsai. *Feature-Based Retinal Image Registration: Algorithms and Validation*. PhD thesis, Rensselaer Polytechnic Institute, Troy, New York, Aug 2003. 9

[223] C.-L. Tsai, A. Majerovics, C. V. Stewart, and B. Roysam. Disease-oriented evaluation of dual-bootstrap retinal image registration. In *Proceedings of the 6th International Conference of Medical Image Computing and Computer-Assisted Intervention (MICCAI 2003)*, volume II, pages 754–761, 2003. 6, 66, 70

[224] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987. 22, 40, 43, 136, 138

[225] T. Tuytelaars and L. V. Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 1(59):61–85, 2004. 2, 166

[226] J. Um and R. Wright. Video strip mosaicking: a two-dimensional approach by convergent image bridging. *International Journal of Remote Sensing*, 20(10):2015, July 1999. 1

[227] K. Umeda, G. Godin, and M. Rioux. Registration of range and color images using gradient constraints and range intensity images. In *International conference on pattern recognition (ICPR'04)*, 2004. 41, 124

[228] M. Uyttendaele, A. Eden, and R. Szeliski. Eliminating ghosting and exposure artifacts in image mosaics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 509–516, 2001. 20, 45, 46

[229] H. Wang, W. Hsu, K. G. Goh, and M. L. Lee. An effective approach to detect lesions in color retinal images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 181–187, 2000. 3

[230] G. Ward. Fast, robust image registration for compositing high dynamic range photographs from handheld exposures. *Journal of Graphics Tools*, 2003. 1

[231] R. Weleber and K. Gregory-Evans. *Retina*, chapter 18. Retinitis Pigmentosa and Allied Disorders, pages 362–460. Mosby, 2001. 13, 68

[232] W. M. Wells III. Statistical approaches to feature-based object recognition. *International Journal of Computer Vision*, 21(1–2):63–98, Jan. 1997. 20

[233] W. M. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis. Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1):35–51, 1996. 32, 34

[234] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, Oct 1992. 40, 43, 138

[235] M. Werman, M. Qiu, S. Banerjee, and S. D. Roy. Robot localization using uncalibrated camera invariants. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 1999. 41

[236] G. Yang, C. V. Stewart, M. Sofka, and C.-L. Tsai. Alignment of challenging image pairs: Refinement and region growing starting from a single keypoint correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007. Accepted for publication. 106, 118

[237] B. H. Yoshimi and P. K. Allen. Alignment using an uncalibrated camera system. *IEEE Transactions on Robotics and Automation*, 11(4):516, 1995. 41

[238] L. Zelnik-Manor and M. Irani. Multi-frame estimation of planar motion. *ijcv*, 22(10):1105–1116, 2000. 38

[239] H. Zhang, P. A. Yushkevich, and J. C. Gee. Registration of diffusion tensor images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 842–847, 2004. 40

[240] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994. 28

[241] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, Nov. 2000. 22, 40, 43, 136, 138

[242] W. Zhao, D. Nister, and S. Hsu. Alignment of continuous video onto 3D point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1305–1318, 2005. 21, 68

[243] L. Zhou, M. Rzeszotarski, L. Singerman, and J. Chokreff. The detection and quantification of retinopathy using digital angiograms. *IEEE Transactions on Medical Imaging*, pages 619 – 626, Dec. 1994. 3

[244] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21:977–1000, 2003. 25, 35, 40

[245] I. Zoghlami, O. Faugeras, and R. Deriche. Using geometric corners to build a 2d mosaic from a set of images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 420–425, 1997. 27