Understanding Image Registration

Towards a Descriptive Language of Computer Vision

by

Steve W. Oldridge

B.Sc., The University of Victoria, 2000
M.Sc., The University of British Columbia, 2002

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

April, 2011

© Steve W. Oldridge 2011

Abstract

Vision researchers have created an incredible range of algorithms and systems to detect, track, recognize, and contextualize objects in a scene, using a myriad of internal models to represent their problem and solution. However in order to effectively make use of these algorithms sophisticated expert knowledge is required to understand and properly utilize the internal models used. Researchers must understand the vision task and the conditions surrounding their problem, and select an appropriate algorithm which will solve the problem most effectively under these constraints.

Within this thesis we present a new taxonomy for the computer vision problem of image registration which organizes the field based on the conditions surrounding the problem. From this taxonomy we derive a model which can be used to describe both the conditions surrounding the problem, as well as the range of acceptable solutions. We then use this model to create testbenches which can directly compare image registration algorithms under specific conditions. A direct evaluation of the problem space allows us to interpret models, automatically selecting appropriate algorithms based on how well they perform on similar problems. This selection of an algorithm based on the conditions of the problem mimics the expert knowledge of vision researchers without requiring any knowledge of image registration algorithms. Further, the model identifies the dimensions of the problem space, allowing us to automatically detect different conditions.

Extending beyond image registration, we propose a general framework of vision designed to make all vision tasks more accessible by providing a model of vision which allows for the description of *what* to do without requiring the specification of *how* the problem is solved. The description of the vision problem itself is represented in such a way that even non-vision experts can understand making the algorithms much more accessible and usable outside of the vision research community.

Preface

Appendix B which formally defines a general model for the field of computer vision was developed primarily by Dr. Gregor Miller, with help from the author, Dr. Sidney Fels and Dr. Daesik Jang. Appendix A which attempts to formally define the problem of image registration was adapted from this general model by the author with help from Dr. Miller.

Similarly, the representation, conditions, and expression of face detection presented in Chapter 8 were adapted from a model created by Dr. Daesik Jang, with support from the author, Dr. Gregor Miller, and Dr. Sidney Fels.

Publications related to this thesis include: [74] (Chapter 4), [72, 73] (Chapter 7), and [40, 64, 65, 90] (Chapter 8)

Table of Contents

List of Tables

List of Figures

Acknowledgements

This work would not have been possible without the vision and determination of Dr. Sidney Fels. By encouraging me to explore the road less traveled we have hopefully shaped the future of computer vision, making it more accessible and available to researchers without a direct vision background. In addition deep gratitude goes to Dr. Gregor Miller who served as a post doc, mentor, colleague, sanity check, and friend throughout the latter half of my thesis. His dedication to this project, his inspiration, not to mention his coding skill, helped make it what it is today. Finally, to the many members of the human communication technologies lab, who came and went as my thesis grew from a concept to reality. Your work inspires me, as does the vision of the lab. In my mind the human factors remain the most important in good engineering and design. Keep fighting the good fight.

Funding for this thesis was provided by NSERC, PRECARN, Bell Labs, Vidigami, and the University of British Columbia. Like thousands of grads before me I am deeply indebted to them.

Dedication

First and foremost, I would like to thank my parents Dennis and Sue Oldridge for their constant encouragement and support. Without their guidance and love I would not be here today.

Meghan, your unconditional love and support has been essential, as has the inspiration that you are in my life. I look forward to continuing this journey together.

To my friends and family who waited long for this moment. Thank you. Your support over the years has been tremendous. I look forward to the new projects and experiences that this achievement will open up to me, and to sharing those experiences with you.

Chapter 1

Introduction

"The eye sees only what the mind is prepared to comprehend."

– Henri Bergson

Vision is an innate part of the human experience, crucial to our ability to comprehend the world around us. One month after birth our eyes begin to focus on objects around us. Within three months we can recognize our parents and siblings. Vision provides a significant source of the information used by our brain to create an internal model, or representation, of the world around us. In order to share our internal model with another, we must come to an agreement about the different aspects of our particular model or representation. This consistency of representation across internal models is essential in order to facilitate the meaningful sharing of information. Our sharing of models with one another most often takes the form of a language.

Computer Vision is the study of how computers and machines see and understand the world. This 'understanding' is similarly achieved: by creating models or representations of a scene. Vision researchers have created an incredible range of algorithms and systems to detect, track, recognize, and contextualize objects in a scene, using a myriad of internal models to represent their problem and solution. In order to effectively make use of these algorithms sophisticated expert knowledge is required to understand, and properly utilize, the internal models used. Researchers must understand the vision task and the conditions surrounding their problem, and select an appropriate algorithm which will solve the problem most effectively under these constraints.

In part, this expert knowledge is required because the predominant model surrounding computer vision problems is algorithm centric, describing and organizing vision problems according to *how* an algorithm solves them. While this type of taxonomy provides an excellent basis for the comparison

amongst researchers of different methods, it does not directly address the conditions surrounding the problems themselves. Information regarding the set of conditions a given algorithm performs best under is difficult to convey without an agreed upon model which represents the problem space itself. Conversely, knowledge of which algorithm best suits a particular set of conditions in the problem space is also difficult to determine without a model to represent those conditions by. For most problem domains computer vision is without a unifying model or representation from which to describe the conditions of the problem and the range of desired solutions.

This thesis presents a new taxonomy for image registration based on the common conditions surrounding the problem. From this taxonomy we derive a model which can be used to describe both the common conditions surrounding the problem, as well as the range of acceptable solutions. In addition to the reasons mentioned above, descriptive methodologies are preferential to purely algorithmic representations in a number of ways. First, they give non-experts access to advanced image processing techniques without requiring specific knowledge of the underlying algorithms that implement them. Second, they allow improved algorithms to seamlessly replace older implementations providing those using a problem centric software library with an instantaneous upgrade path, without reprogramming or integrating a new implementation. Finally, if as often happens the conditions around the problem change, the programmer can automatically select a more appropriate algorithm simply by changing their description.

Extending beyond image registration, our proposed general framework of vision is designed to make all vision tasks more accessible to developers, by providing a model of vision which allows for the description of *what* the developer wants to achieve, without requiring the specification of *how* the problem is solved. In order to provide this accessibility a common representation for the significant conditions surrounding a given vision problem must be established, and the solution space must be well defined. This is a difficult task, and requires an in depth understanding of the field, however once established, vision researchers who are developing new algorithms see significant benefits to the widespread understanding of this knowledge. First, they can identify and represent the conditions under which their algorithms perform well, allowing for much more robust comparison of algorithms. The existence of a problem space which models all of the possible conditions under which a vision task may be performed allows for the creation of test sets which span well defined problem conditions, allowing for a more direct comparison of performance. Second, by examining the problem space and the existing algorithms that support it researchers can identify niches within

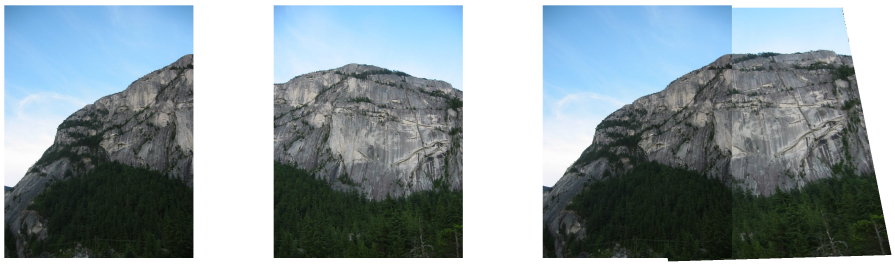

Figure 1.1: Input images taken from a panorama, and the corresponding rendered image derived by stitching the aligned pair using a feature based method.

the problem space which are useful but do not have solutions. Finally, the description of the vision problem itself is represented in such a way that even non-vision experts can understand making the algorithms much more accessible and usable outside of the vision research community.

Reorganizing computer vision in this way requires a deep understanding of individual vision problems. Within this thesis we have focused on image registration problems, providing a starting point for our proposed descriptive language model of vision, OpenVL. Image registration was chosen because it is a mature problem with a wide range of solutions that work well under specific conditions. This exploration provides a pathway for further development of our language through expansion into other areas of vision.

1.1 Understanding Image Registration

Image registration is the process of calculating spatial transforms which align a set of images to a common observational frame of reference, often one of the images in the set. Registration is a key step in any image analysis or understanding task where different sources of data must be combined. It is a critical component of computational photography [2], remote sensing [16, 49], automated manufacturing processes, and medical image processing [55, 80]. More recently it has been used to create navigable models of a scene from a database of photographs [91], to remove unwanted objects from overlapping images, and in video stabalization. Figure 1.1 shows a pair of images, and the corresponding image that can be created by properly registering them.

Seminal surveys of registration by Brown [14], Zitová and Flusser ,[103]

and most recently by Szeliski [93] all divide the field algorithmically, focussing on *how* registration is accomplished. While this approach provides a good basis for classifying and comparing algorithmic similarity, it does little to illuminate the problem of registration itself, particularly to programmers who are unfamiliar with specific registration techniques and their applicability.

When the images vary by more than just alignment the proper selection of appropriate algorithm is critical in calculating the correct spatial transform. Figure 1.2 introduces some common types of image registration problems: panoramas, focal stacks, high-dynamic-range images, and finally super-resolution images. Both focal stacks and high-dynamic-range images vary in parameters other than simply alignment.

Several important image registration techniques and strategies have been developed since Zitová and Flusser first published the most recent survey. The increased computational power of the past decade has made automatic methods the norm, and under optimal conditions modern algorithms are able to align image pairs more accurately than can be detected by the human eye [5]. In Chapter 2 we organize these new methods according to the traditional algorithm centric taxonomy, providing an up to date survey of image registration that forms the basis for our understanding of the field. This algorithm-centric taxonomy reflects the current means by which researchers must determine the appropriate registration algorithm for their particular problem.

From this understanding of algorithms, a new taxonomy is proposed in Chapter 3, which divides the field of image registration into taxa based on the conditions of the problem being solved. By rethinking the problem in this way we change the abstraction from one requiring knowledge and expertise about particular algorithms, both in how they work and when to use them, to one requiring expertise about the registration problem itself. A significant advantage of this problem centric methodology is that the conditions surrounding the image registration problem rarely change, whereas new algorithms for image registration are constantly being developed. This can cause problems for non-experts in determining which of these algorithms is most appropriate because the knowledge of algorithmic appropriateness is empirical and constantly changes as new algorithms are developed. Within our taxonomy image pairs are categorized according to the categories: non-varying, intensity varying, focus varying, sensor varying, and structure varying based on their reported performance in the literature.

Within the literature algorithms and systems are most often described as useful for one particular application area. In most cases, however, the meth-

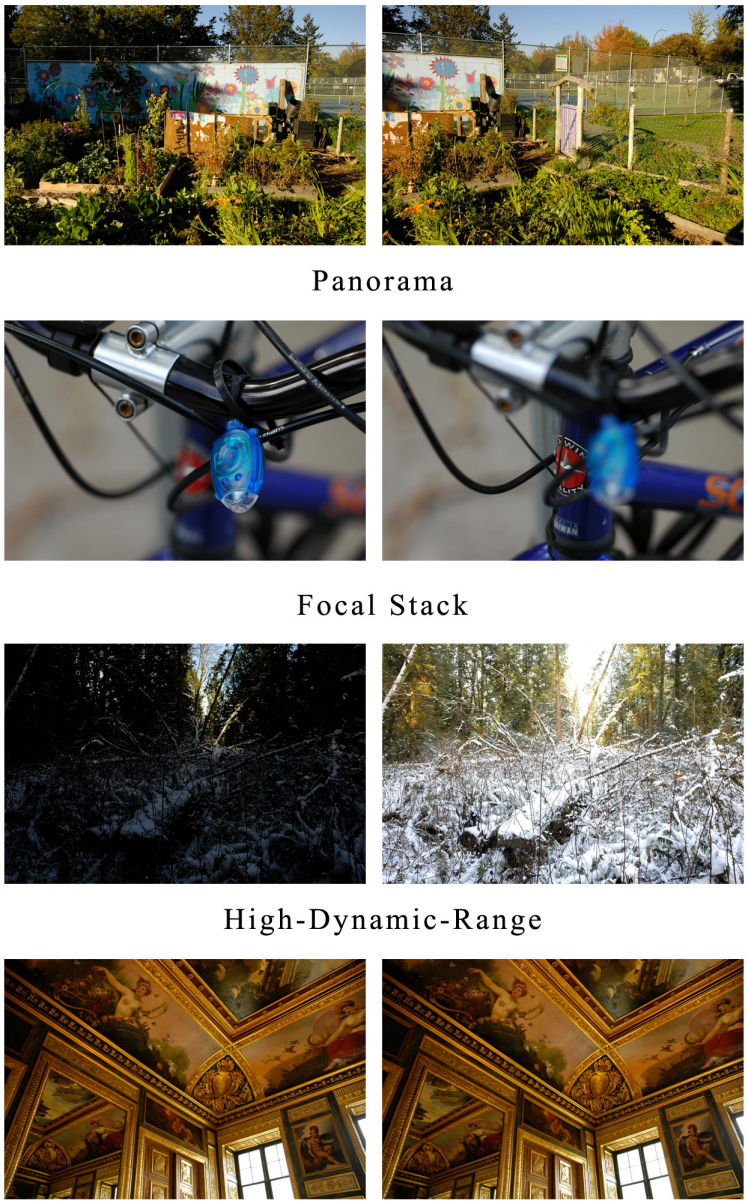

Figure 1.2: Image pairs representative of image registration problems common to computational photography. Determining the appropriate registration algorithm under each set of conditions currently requires extensive research into the literature.

ods used in these applications can be applied to a limited subset of problems from other applications. This distinction between image registration *methods* and image registration *problems* is important. Binary classification of a problem type does not allow for the level of distinction required to know how effective a given algorithm will be within a particular range of the problem space. Instead, we utilize these forms of variation as the major dimensions of the image registration problem space, forming the basis for a model of registration which we present in Chapter 4. Using a formal definition of image registration and a well defined model of the conditions surrounding the image registration problem space and the array of possible solutions we create a mechanism by which vision researchers can define both the conditions and solutions that their algorithm supports, and the conditions and desired range of solutions of particular instances of an image registration problem. Within our model, individual image registration algorithms can be thought of as supporting a volume of solutions across the entire space. Similarly the common types of image registration problems can also be thought of as occupying a volume. By decoupling algorithms from common types of image registration problems through this mapping, we gain insight into the comparative performance of algorithms and an understanding of where algorithms support a problem space well, and what areas are not well supported.

This common model of the problem and solution space is critical in order to be able to compare algorithms in a well defined way. Chapter 5 outlines a methodology for creating image registration testbenches which span the problem space, allowing for the direct comparison of algorithms. Three testbenches of 5000 image pairs which explore different volumes of the problem space are then created and an analysis of how well each algorithm performs in the face of variation in transform parameters, image size, overlap between images, exposure, and focus is performed. Analysis of variance techniques are then used to analyze which of the testbench parameters affect the quality and likelihood of a solution.

Combining the mappings of algorithm and problem specification with our understanding of how well different algorithms perform in the different areas of the image registration problem space, our model can then interpreted and appropriate algorithms can be selected based on the problem representation and conditions. The interpretation of the model of the image registration problem, and the selection of appropriate algorithms is discussed in Chapter 6. Our proof of concept interpreter is based on our testbench for image registration problems using a direct evaluation of the performance of image registration algorithms under similar problem conditions.

In addition, when the conditions surrounding image registration prob-

lems can be detected it becomes possible to create a system that is able to automatically populate the model, and select the most appropriate image registration algorithm to align a pair of images purely from the image pair themselves. In Chapter 7 we explore two such methods. First a simple rule based expert system is examined, which provides a one to one classification of whether it expects a given image pair to be appropriate for a variety of algorithms. A second system was developed which uses support vector machines to classify between panoramas, high-dynamic-range images, focal stacks, super-resolution, and unrelated image pairs. Each of these types of image registration problems can be thought of as representing a volume clustered around a point in our n-dimensional problem space. By classifying the type of registration problem and choosing an appropriate method the system significantly improves the flexibility and accuracy of automatic registration techniques.

1.2 Towards A Descriptive Language of Computer Vision

Expanding upon the methodology explored through our creation of a model for image registration, we propose the development of a general model of computer vision. This project is the work of several researchers, who are collaborating to develop the Open Vision Language (*OpenVL*) which we introduce in Chapter 4 for image registration, and generalize in Chapter 8. The goal of OpenVL is to provide a similar abstraction over the algorithmic and implementation specific aspects of vision by providing a language capable of expressing vision problems. The proposed abstraction supports code reuse, hardware acceleration, exploitation of advanced techniques and extensions to the language or algorithms.

One of the significant problems preventing widespread adoption of computer vision is the lack of a framework that separates the need for knowledge of a vision concept from knowledge of specific vision algorithms. There have been many attempts to create open repositories of software supporting the vision community [12, 15, 32, 67, 99], however they provide vision components and algorithms without any context of how these may be applied. In order to implement sophisticated systems users of these libraries still need expert vision knowledge, both of which algorithms and settings are applied to solve particular problems, and also of how the components and algorithms combine. As we found with image registration, the knowledge surrounding vision problems as a whole is largely empirical.

By way of example, a widely used computer vision library, OpenCV [12], provides all of the components necessary to implement face detection in several ways, and even includes an example of how this can be done using Haar descriptors. In order to modify the example in a meaningful way however a developer implementing a face detector needs some knowledge about Haar descriptors. If this implementation doesn't fit their problem, for example if their problems contain more occlusion than the OpenCV algorithm can handle, another detection algorithm must be researched, implemented, and integrated until one that fits their particular problem has been solved. By implementing several algorithms the developer may begin to gain some understanding of the problem of face detection, however their significant efforts have made them somewhat expert in the field. More likely a developer with no vision experience will simply accept the available implementation as is and will make do with the limited functionality or performance.

Instead we propose to illuminate the common issues and context of vision problems directly in the problem model itself, providing developers with a direct understanding of the different types of conditions and tradeoffs that necessitate the use of particular algorithms or settings, without requiring knowledge of all of the algorithms capable of the task. If they understand these concepts and can describe their problem sufficiently well within a suitable framework, then an interpreter should be able to infer from the description which algorithm to use. This strategy leverages the expert knowledge of vision researchers, who better understand the problem domain and the various algorithms that are used to solve within it, while simultaneously increasing the knowledge of important vision concepts amongst non-experts. Although this creates a group of vision developers who do not understand how underlying algorithms are solving their problem, we would argue that this is a common and significant advancement in any field. Few individuals understand how compression algorithms work, but a wide range of individuals, including photographers and artists use it extensively. In compression the important concepts from a usability perspective can be broken down into the quality and size of the image, a tradeoff that non-compression experts can understand and make use of. While these models of the problem space, and their corresponding mapping into vision algorithms do not necessarily exist for all vision problems, they certainly exist for many, and would make computer vision accessible to a much wider audience if properly implemented. This is one of the main goals of OpenVL.

Our desire with OpenVL is to provide access to sophisticated computer vision algorithms and datatypes through descriptions of the problem. The OpenVL interface is descriptive, unlike the usual procedural interface which

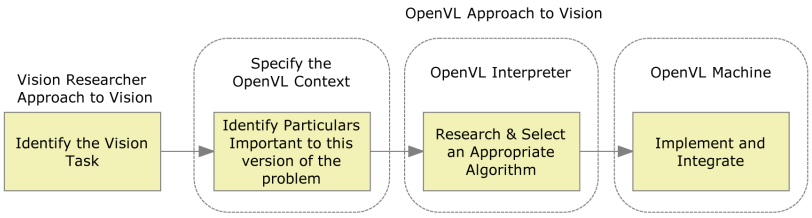

Figure 1.3: OpenVL mimics the approach taken by experienced vision researchers, providing a context within which non-experts can describe aspects of their problem which are important in the selection of an appropriate algorithm. The OpenVL machine can then interpret the context, selecting one of the already implemented algorithms designed to solve this particular part of the problem domain.

is specific to algorithms and data structures. A programmer specifies *what* it is they want to do, instead of *how* they want something done, which provides an abstraction layer above algorithmic details. Our interface allows the application programmer to specify what type of scene their data represents and the result they require. This is consistent with overcoming the usability problems associated with image understanding as discussed in the RADIUS project [27], which used manipulable 2D/3D models of the scene to help guide the choice of image processing algorithms for non vision experts. Likewise, the Image Understanding Environment Project (IUE) [67] attempted to provide high level access to image understanding algorithms in order to make them accessible and easier to reuse, however this approach was algorithm centric. Differentiation of our proposed framework from these systems is explored in Section 2.2.

Much of our motivation comes from the success of similar endeavors in computer graphics. Frameworks such as OpenGL and DirectX provide an abstraction which offers the use of advanced graphics techniques to a large user base, not just specialists in graphics. The abstraction gives programmers access to conceptual processes (e.g. drawing a textured triangle) but hides the complexity of the operation from the programmer. These frameworks have been designed to work on different hardware without requiring additional effort by the application programmer, providing a model for similar acceleration in vision. However, graphics is a structured problem and is more easily represented in both concept and hardware. The image under-

standing problem is unstructured and requires a higher level of sophistication to extract meaningful models. Our language model attempts to bring control of sophisticated concepts such as image registration, tracking, and even object recognition onto a level similar to the drawing of a simple shape in graphics.

OpenVL gives programmers who are not vision specialists access to advanced processing techniques without specific knowledge of the underlying algorithms that implement them. Algorithms are provided by specialized vision researchers who integrate their work by evaluating or describing the conditions under which their technique operates well. In order to make this interpretive leap developers must be able to identify where within the problem space their particular version of the problem lies. Similarly, vision researchers developing algorithms to be used within OpenVL must be able to map the performance of their algorithm in the different parts of the n-dimensional problem space. The proper identification and development of a problem model is critical to this process and is quite a difficult undertaking, however this process illuminates the problem space in a way that incremental research and advancement fails to, and can provide insight and direction for the computer vision community. Integration of algorithms is discussed in more detail Section 8.2.

'Expert' vision researchers also stand to gain significantly from the use of OpenVL. Vision research commonly relies on well established vision components and algorithms as precursors or parts of an overall system. The choice of component algorithms can have a significant impact on the overall system, affecting the performance of novel components of the research significantly. Specialist researchers are often experts in one or more vision problems, but rarely understand all areas of vision. The development efforts required to explore all of the various algorithms and options available and their impact on the novel portions of the research is significant enough that most researchers are forced to forgo this level of investigation and work with algorithms that are available. In addition the comparison of novel components or algorithms to existing methods is often another difficult aspect of research. Without models to define the space of a particular vision problem, the examples used in the comparison of algorithms may not be equivalent or representative of the whole problem space. Through the use of common model based representations of problems and testbenches, such as those seen in Chapter 6, OpenVL provides a mechanism for researchers to directly compare their novel component across the entire problem space.

The abstraction OpenVL provides over algorithms has several advantages for the vision community. If application developers use OpenVL to

describe their problem then the code written can be reused with newer implementations of OpenVL, utilizing the latest computer vision algorithms as research evolves. Another advantage is the separation of hardware implementations, which allows continued improvement of performance through new acceleration technologies without the need to recompile or target applications to a specific device. We firmly believe that an abstraction which separates application developers from specific algorithms and supports hardware acceleration will allow research in computer vision techniques to advance more quickly and be more easily shared.

1.3 Contributions

The contributions of this thesis are as follows:

- First, an up to date survey of image registration techniques was carried out using an existing taxonomy of image registration.
- From this initial mapping a new taxonomy was developed and existing techniques were again mapped according to the new taxa.
- A model of the image registration problem space was developed based on this taxonomy, allowing vision researchers and non experts to describe the representation and conditions of image registration problems in a well defined manner.
- A method for automatically creating testbenches was developed for the image registration problem space.
- Three testbenches of 5K image pairs were created and tested, with five image registration methods evaluated for performance under the set of conditions that each testbench encompassed.
- A method of interpretation was developed which allows for the selection of appropriate algorithm(s) based on a given problem space description.
- Two methods of automatic detection of common types of image registration problems were developed, allowing for the automatic classification of image registration problems.
- Finally, the methodology used to create a model for image registration was extrapolated providing a starting point for a general model of computer vision.

Publications related to this thesis include: [40, 72–75, 90]

Chapter 2

Related Work

"Every man takes the limits of his own field of vision for the limits of the world."

– Arthur Schopenhauer

In this chapter we perform a detailed examination of image registration algorithms and existing frameworks. This is an important step in understanding the problem space of image registration. Surveys of image registration [14, 93, 103] divide the field algorithmically, focussing on *how* registration is accomplished. This approach provides a good basis for classifying and comparing algorithmic similarity. These models also begin to illuminate the conditions important to the problem of registration, guiding the creation of our novel taxonomy in Chapter 3.

Several important image registration techniques and strategies have been developed since the most recent image registration survey. The increased computational power of the past decade has made automatic methods common, and under optimal conditions modern algorithms are able to align image pairs more accurately than can be detected by the human eye [5]. Following the most recent taxonomization [103] we have differentiated these new methods into the categories of area based, feature based methods, mutual information based methods, and non-rigid methods. As with Zitová and Flusser we do not examine details of particular algorithms or perform comparative experiments, but instead attempt to map and summarize the main approaches used in registration today. The direct comparison of image registration algorithms is a difficult concept rarely addressed in literature. Section 2.1.4 explores this concept in more detail.

To support our goal of a general model of computer vision we examine both past and current attempts at the development of computer vision libraries and frameworks. Early attempts at vision such as automatic image

understanding had similar goals to OpenVL in that they were an attempt to make vision more accessible to non-vision experts. Unfortunately complete automation of vision tasks has thus far proven unfeasible. Conversely, most current vision libraries provide extensive functionality without any inference or context about how individual algorithms may be used. OpenVL seeks a middle ground between these extremes, and as such the examination of examples of each type of vision frameworks is critical to its success. Visual programming environments were an attempt at simplifying vision tasks by representing them through data-flow maps, although significant knowledge of algorithms was still necessary, and are worth examining. Finally, previous instances of declarative languages for image processing provide insight into how a language can be structured so that the application developer is describing *what* the program should accomplish, rather than describing *how* to go about accomplishing it, one of the main tenants of OpenVL.

2.1 Image Registration

Past image registration surveys provide a methodological taxonomy for understanding the different algorithms used to solve the registration problem. Brown [14] divides registration into four components: feature space, search space, search strategy, and similarity metric. The later work of Zitová and Flusser [103] divides the field into area and feature based methods, and their model reflects the shift towards feature based methods that occurred between the two papers. The four basic steps of image registration under their model are: feature detection, feature matching, mapping function design, and image transformation and resampling. Like Brown we have chosen to leave image transformation and resampling out of our taxonomy; these steps, though important for applications involving image registration, are rendering problems, and are independent of spatial alignment. Szeliski [93] similarly divides the field into direct (pixel) based registration and feature based registration. Maintz [55] provides insight into the use of registration in medical imaging, providing important methods and variations relevant to that field. The taxonomy divides both algorithmically and based on the modality of the data, again providing a similar mapping. Pluim et al. [80] also survey medical imaging, focusing on Mutual-Information-Based registration techniques. Their taxonomy classifies algorithms into two main categories: methodological aspects and aspects of application.

Although the field is rapidly moving towards automatic image registration, algorithms and systems are most often limited to a single applica-

tion area such as stitching panoramas, super-resolution, high-dynamic-range (HDR) imaging, focal stacking, multimodal imaging, etc. In most cases the methods used in these applications can be used on a limited subset of problems from other applications. This distinction between image registration *methods* and image registration *problems* becomes important in Chapter 3 as we explore our new taxonomy of image registration. It is important to note that no single algorithm exists that will solve all types of registration problems.

Brown's framework outlines how knowledge of the types of variation that occur in image sets can be used to guide selection of the most suitable components for a specific problem. Variations are divided into three classes: variations due to differences in acquisition that cause the images to be misaligned, variations due to differences in acquisition that cannot be easily modeled (such as lighting or camera extrinsics), and finally variations due to movement of objects within the scene. These are labeled by Brown "corrected distortions", "uncorrected distortions", and "variations of interest" respectively. Zitová and Flusser provide a model of variation according to the manner in which the images were acquired: different viewpoints, different times / conditions, different sensors, and finally scene to model registration. Within their survey they do not use this mapping directly, however in many cases they discuss the type of problem each method has been designed to solve, allowing a similar mapping of methodology from situation. Similarly Pluim et al.'s "aspects of the application" entail image modalities, subject of the registration, and the object of registration. This delineation provides an excellent starting point for variations that are important within the medical imaging community. As we will see in Chapter 3 it is this concept of variations that we have chosen to base our taxonomy on, extending these initial ideas into specific variations common in image registration and exploring the successful algorithms under different problem conditions.

As with Zitová and Flusser we do not examine details of particular algorithms or perform comparative experiments here, but instead attempt to map and summarize the main approaches used in registration today. It would be impossible to provide an evaluation of every image registration algorithm, or even every type of registration problem. In Chapter 6 we introduce a testbench which allows for the evaluation of a variety of image registration problems. Finally, although we include multimodal image registration in our taxonomy, there is not space to cover the whole of medical image registration; the field is so vast and has grown so significantly over the same period that an entire survey and taxonomy could easily be devoted

to that subject alone.

2.1.1 Area-Based Methods

The first attempt at image registration was performed by an area based method developed by Lucas and Kanade [53]. Area based methods work by comparing some measure of the aligned pixel values as an error function to be minimized. Typically a sum of square difference of the pixel intensity is used. The process proposed in [53] is an iterative gradient descent: at each iteration calculate the current error, and using the slope of the error space at that point and an estimate of the Jacobian calculate the next position.

This method is susceptible to local minima, but works reasonably well for images with a similar intensity, particularly when applied at multiple scales via image pyramids, and assuming the spatial overlap is significant. In the case of focus stacks it is particularly successful, often outperforming feature-based methods which cannot find features in the same location across images. Significant spatial overlap is often the case for focus stacking problems, particularly those composed of microscope data where sensor movement is minimal between images. Bradley et al. [11] make use of normalized cross correlation in their virtual microscopy system requiring an overlap of at least 45% between image pairs and ignoring results that fall outside their expected solution area.

Extending beyond direct metrics of intensity, Schechner and Nayer [87] presented an alignment method based on pyramids of maximum likelihood as a part of their approach to generalize panorama images to incorporate HDR. According to the authors, the addition of uncertainty into the intensity based search space allows for a 'better' alignment under these conditions. Figure 2.1 shows the corresponding mosaic regions of an unaligned, intensity aligned, and maximum likely aligned set of images.

Ward [82, 100] introduced a method specifically designed to align images with significant variations in intensity. The technique thresholds image pairs into pyramidal bitmaps, creating binary images that represent regions that are neither over nor underexposed. The bitmaps are analyzed and aligned for translation errors using shift and difference operations at each level of the pyramid. With this method 3 megapixal image sets are aligned in a fraction of a second. Unfortunately their method deals solely with translation errors, although they discuss the possibility of solving for rotation errors, suggesting that 10% of their data set failed as a result. This binary 'pass' / 'fail' evaluation of the registrations is indicative of the poor evaluation techniques used by researchers in the field. Section 2.1.4 goes into this problem in

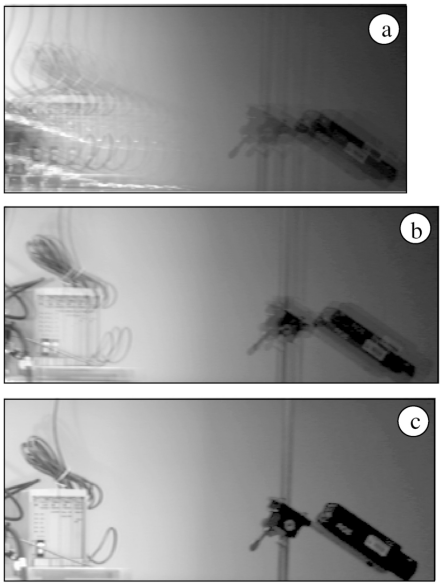

Figure 2.1: Corresponding mosaic regions. (a) Attempting to register the raw images by minimizing their squared difference fails. (b) Compensating for the spatially varying attenuation without accounting for uncertainty amplification does not lead to a good registration. (c) Accounting for spatially varying uncertainties due to the attenuation compensation leads to successful image registration. [87]

more detail. In addition, a method that performs two dimensional search of what would traditionally be a 6d (affine) or 8d (perspective) solution space leads the author to believe that their data set was somewhat tailored to the algorithm. Still, the method remains one of the more successful instances of intensity invariant registration to date indicating the difficulty level of the problem.

Sharma and Paval [89] apply traditional area based techniques to overcome differences in intensity and gradient common in multimodal images by making the images as similar as possible, transforming images into representations invariant to polarity reversals. Irani and Anandan [37] similarly transform images into high-pass energy images which are significantly less sensitive to sensor variations. These methods have been further developed by Liu et al. [50, 51] who use Gabor filtering as their local frequency representation. More recently, Henn and Witsch [34] define two nonlinear distance functions and minimize these to find the optimal alignment.

In the field of medical imaging, maximization of mutual information, developed simultaneously by Viola and Wells [98] and by Collignon et al. [54], is another area based method which uses an error function other than differences of intensity. Successive methods were comprehensively surveyed by Pluim, Maintz and Viergever [80]. Bardera, Feixas and Boada [7] proposed two new similarity measures based off of Jensen's difference applied to Rẽnyi and Tsallis-Havrda-Charvãt entropies claiming that their proposed measures are more robust than the normalized mutual information for some modalities and a determined range of the entropy parameter. Gan et al. [31] suggest using Kullback-Leibler distance if a priori knowledge of the joint intensity distribution is available. Makela et al. [57] provide an overview of further methods focusing specifically on cardiac images.

2.1.2 Feature Based Methods

When dealing with images which only vary spatially feature based methods [61] are the most common technique applied, and are generally faster and considered more accurate unless the image pairs contain little high-frequency information from which to find and match features.

Feature based methods work by matching a similar set of features across two images. Different strategies are available for the selection of feature sites, known as 'interest regions,' however features are most commonly selected at points of interest throughout the image, usually using some sort of corner detector. Adaptive non-maximal suppression [61], among other strategies, has been used with significant success to prevent the selector from picking all of its points in the same spot in the image, helping in instances where the images contain busy regions that are non overlapping. An implicit assumption in the selection process is that these sites of interest will occur in the same place in both images. While this is be true for some images, it is not necessarily the case when intensity, focus, sensor or structure change. When this occurs the feature sets of the two images are likely to be different, and matching is difficult at best. Some methods attempt to solve specific instances of this problem by selecting feature sites that are meant to be invariant across one or more of these dimensions, however few of these are successful.

A hotly debated and active area of research in feature based image registration is the choice of feature descriptor. The descriptors must be distinctive and at the same time robust to changes in viewing conditions as well as to errors of the detector. The comparison of different feature descriptors has been recently made by Mikolajczyk and Schmid [63]. In their survey they

compare shape context [8], steerable filters [30], PCA-SIFT [43], differential invariants [45], spin images [48], SIFT [52], complex filters [86], moment invariants [97], and cross-correlation for different types of interest regions, concluding that the ranking of the descriptors is mostly independent of the interest region detector and that the SIFT-based descriptors perform best.

While features based on mutual information have been used in pattern recognition [78, 95] they have not to our knowledge been used successfully for registration of multimodal images.

Sand and Teller [85] attempt to handle registration of image pairs for high-dynamic-range images by only selecting features from parts of the image that can be more easily matched while avoiding parts that are difficult. A combination of area based methods and feature based methods are used to iteratively align the images, ignoring feature points that occur in over or under exposed regions of the image. Their technique was designed for matching two video sequences, reporting good results with the limited spatial variation that entails, however it was not tested on still photographs. More recently Tomaszewska and Mantiuk [94] presented a similar idea, reporting a high quality alignment such that the "photographs were aligned with sufficient accuracy so that there are no visible artifacts in the final HDR image" by using only features that occur across all images in the set. These methods of culling features that are inappropriate only work when enough features remain to make a proper alignment.

Once features have been detected they must be matched across images. Approximate nearest neighbor matching using k-d trees is the dominant method of matching and is widely used by a number of researchers [61]. Exact nearest neighbor approaches are computationally intractable for large numbers of features and dimensions, and provide a limited advantage over approximate methods [58].

Once each feature has been matched to a corresponding feature in the other image, alignment is found by solving the overconstrained system created by the feature pairs. The ideal solution should minimize the distance between features once the two images have been aligned. Random Sample and Consensus (RANSAC) is commonly used [61] because of its robustness and efficiency, however other solvers such as nonlinear least squares fitting have also been applied with success. Yang et al. propose using an Iterative Closest Point Dual Bootstrap method [102] which was reported to perform favorably to RANSAC for difficult image pairs.

2.1.3 Non-Rigid Methods

Image pairs can vary significantly in the structure of the scene they depict, either because objects within the scene have moved, or more commonly in medical imaging because objects have changed over time. Aligning images in spite of these changes often requires non-rigid transforms that solve for the alignment of regions or at the extreme of individual pixels. The selection of points on the grid of solutions, and interpolation of values between those points, along with the algorithms used to solve for the global and local solutions vary from algorithm to algorithm and are the main differentiating factors within this type of registration.

Bookstein proposed one of the seminal concepts of non-rigid models for image registration: using thin plate splines to interpolate affine transforms [9]. Moshfeghi proposed an alternative model based on elasticity [66]. Christensen et al. introduce the idea of viscous fluid representations of deformable registration [19], while Bro-Nielson and Gramkow [13] significantly accelerate this concept using a fast fluid model. McInerney and Terzopoulos [62] provide a good survey of early non-rigid techniques and their application within the medical imaging community.

Rueckert et al. [84] present a non-rigid method that uses a global affine transform, followed by a local B-spline matching of normalized mutual information voxels, applying their technique to the registration of breast MR images. Rohde et al. [83] make several contributions to the field: use of radially symmetric basis functions rather than B-splines to model the deformation field; a metric to identify regions that are poorly registered and over which the transformation needs to be improved; partitioning of the global registration problem into several smaller ones; and a new constraint scheme that allows them to produce transformations that are topologically correct. They compare the proposed approach to more traditional ones listed above and show that their new algorithm compares favorably to those in current use. More recently D'Agostino et al. [22] propose modeling the registration as a viscous fluid that deforms under the influence of forces derived from the gradient of the mutual information registration criterion, validating their method by matching simulated T1-T1 weighted magnetic resonance imaging (MRI) images, T1-T2 weighted MRI and T1 weighted to Proton Density (PD) MRI images.

Kang et al. [42] describe a technique for creating high dynamic range video from a sequence of alternating variable intensity exposures. Their sophisticated HDR stitching process uses local area based alignment and non-rigid motion estimation to compensate for camera movement and object

motion within the scene, a technique tailored to their input data.

Crum, Hartkens and Hill present a more recent survey of nonrigid image registration [21] providing a more in depth analysis of this subject.

2.1.4 Evaluation of Registration Accuracy

In order to be able to compare results from different registration methods an evaluation of the accuracy of the registration is necessary. Some registration methods minimize error functions which they hope are representative of the actual error in registration, however this mapping is not always accurate. For example the sum of square difference of intensity used by Lucas and Kanade [53] does not represent the actual error of registration when aligning images which vary significantly in intensity or modality.

Zitová and Flusser [103] identify three measures of registration accuracy in their survey that are independent of error space: localization error, matching error, and alignment error. Localization error represents mistakes in the location of feature based methods' interest regions, usually expressed as the average distance in pixels. Matching error is measured as the number of false matches between features, and is another measure of performance which can only be used to evaluate feature based methods. Finally, alignment error measures the difference between the proposed alignment and the actual between-image geometric distortion. Alignment error is ultimately what researchers must be concerned with when comparing across all possible methods, however this requires ground truth information about the transform which aligns the images.

This ground truth transform is seldom used in the evaluation of image registration algorithms both because it is difficult to create an image pair with a known transform which conforms to particular conditions of a problem, and because image pairs which conform to these conditions but whose transform is unknown are relatively easy to obtain. Each of the image registration algorithms presented in Chapter 2 were evaluated using image pairs that were created by the researchers, and most are evaluated in a visual manner, examining for artifacts or misalignments manually. The comparison of such evaluations is difficult even when researchers directly compare their algorithm to another using the same images since the images chosen could play a role in performance. Examples of this have been included throughout the related work section to give the reader a sense of the type of performance reporting common in the field today.

Azzari et al. [5] recently propose the use of a set of synthetic data rendered using image interpolation techniques and computer graphics tools.

Their "Virtual Camera" simulates the geometric image formation process, taking into account internal parameters, pose and position, sensor size and resolution, focal length, and sensor noise. Their system creates a chain of images with a known homography of ground truth transforms which they have made available online. This method provides an excellent starting point for a testbench of image registration techniques, however their image sets are low resolution (320×240), limited to translation and rotation, and contain no variation in intensity, focus, sensor modality, or scene structure.

One of the key limitations on the resolution of the synthetic images is due to the interpolation of the reference image under the ground truth transform, an issue which Azzari calls the '*pixelation effect*.' In order to avoid pixelation *a minimum distance and maximum rotation of the virtual camera with respect to the scene, given the texture resolution, are estimated beforehand and used as thresholds.* Presumably these thresholds were chosen to limit the possibility that a pixel in the synthetic image was being interpolated without enough available data.

To evaluate the performance of algorithms with this ground truth dataset Azzari suggests three performance metrics. First, the mean square error (MSE) of intensity values is used, which is problematic in instances where intensity, sensor modality, or structural variations have taken place. As we saw above MSE of intensity cannot be used directly to evaluate all types of image registration. Second, they measure the average geometric distance of a grid of control points, calculating the *alignment error* of the transformed points in comparison to the ground truth as suggested in [103]. Finally, they calculate the "number of misplaced pixels," a measure which calculates the number of missing and redundant pixels for a given transform and normalizes across image size. This method involves a thresholding at the pixel level, and "is often a very small number which must be scaled by 10^3", calling its usefulness into question.

A much more robust and high resolution set is necessary for evaluation of modern registration algorithms, and is explored within this thesis in Chapter 5. The creation of such a test set allows for a much more detailed mapping of the image registration problem space. By creating image pairs with both ground truth transforms and known variations in intensity, sensor modality, structure, etc. the performance of algorithms in different parts of the image registration problem space can be measured.

2.1.5 Automatic Classification

Chapter 7 introduces two methods of automatic classification based on detection of the types of variation presented above. Other systems for automatic image registration exist, however they are limited to single application domains such as stitching panoramas [61], super-resolution [29, 104], high dynamic range (HDR) imaging [87, 100], or focal stacking [2]. These techniques can be used on a limited subset of problems from other domains, however no single algorithm exists that will solve all types of registration. Yang et. al [102] extend the flexibility of their algorithm within other problem domains by analyzing the input image pairs and setting parameters accordingly, however the single underlying algorithm still fails in a number of their test cases.

Drozd et. al [25] proposed the creation of an expert system based tool for autonomous registration of remote sensing data, and outline a plan to use information derived from image metadata and user tags to select from amongst correlation based, mutual information based, feature based, and wavelet based methods. Unfortunately their description is more of a preliminary proposal and doesn't provide results of the performance of their expert system or of how appropriate the registration techniques selected were at solving the problems they were chosen for. To our knowledge no other attempts at classifying registration have been made, either by rule based systems or by learning methods.

2.2 Computer Vision Libraries and Frameworks

Many attempts have been made to develop computer vision or image processing frameworks that support rapid development of vision applications. Image Understanding systems attempted to make use of developments in artificial intelligence to automate much of the vision pipeline. Visual programming languages that allow the creation of vision applications by connecting components in a data flow structure were another important attempt to simplify vision development. Medical image processing libraries have been developed which support a community of developers whose knowledge of computer vision is not a given. Declarative programming languages also represent another attempt to provide vision functionality to non vision experts. Finally, open source computer vision libraries or sdks that provide common vision functionality have been critical in providing a base of knowledge from which many vision applications have been developed. These three approaches to vision are examined in detail below.

In addition to providing components and implementations of common vision algorithms, the currently available libraries also provide a number of supporting features that are useful in the development of vision applications but often are operating system, data format, or network specific. This combination of features suggests that these frameworks suffer from a lack of sufficient conceptual organization of the vision problem's constituent tasks. Makarenko et. al [56] demonstrate that lack of scope definition and overlap across frameworks leads to a breakdown in component reusability, suggesting that proper isolation of different components could prove both insightful and useful.

Reexamining the elements necessary for vision application development we propose the following classification of scope for computer vision:

Access	:	Retrieval of data
Transfer	:	Mediation between devices
Convert	:	Conversion into required format
Modify	:	Applying filters, crop, transforms, etc.
Analyze	:	Using vision to model a scene

Decomposing the problem in this way promotes code re-use as well as focussing development effort on well-defined parts of computer vision. Under this classification we focus on the *analysis* problem. The other components in the computer vision pipeline have various example solutions, such as Quicktime7^{TM} for access, Hive[1] or snBench[71] for transfer, ImageMagick

[36] or IUE Data Exchange [46] for conversion and CoreImage™ [3] for modification.

2.2.1 Image Understanding

Image Understanding (IU) represents an important early approach to computer vision frameworks. The goal of image understanding systems is to interpret images by locating, characterizing, and recognizing objects and other features in the scene. This convergence of Artificial Intelligence (AI) and Image Processing techniques was popular in the mid to late nineties. While few of the frameworks or libraries are still in use today, they are examined in order to understand what was being attempted, what was achieved, and what lessons can be learned from their successes and failures. Although our general model of computer vision presented in Chapter 8 is not an image understanding framework, it relies on similar techniques in order to infer the best method solution, using the developer's description of the problem rather than attempting to automatically interpret the scene. As we will see in Chapter 7 image understanding methodologies can also be applied to supplement or even replace the problem and scene description.

The Image Understanding Environment

The Image Understanding Environment (IUE) [46] was an early attempt at creating a unified image understanding library. The goal of the IUE was to support research productivity through a standard object-oriented interface, supporting technology transfer via a platform for demonstrating the benefits of IU algorithms in the context of user applications. Education and development were encouraged by providing standardized formats for encoding algorithms, and computational models were developed that provided the basic data representations and associated operations that all IU architectures must support. Funded by ARPA from 1991-96 and consisting of a consortium of ten companies with a vested interest in vision, an extensive set of functionality was developed. Figure 2.2 outlines the major partitions of the IUE.

Functionality was intended to include Image Enhancement, Shape from X, Stereo, Motion, Colour, Surface Organization, Edge Detection, Photogrammetry, Object Recognition, Model Matching, Perceptual Grouping, Texture Analysis, Image Based Modelling, and Region Segmentation, providing it with a robust collection. Additionally, the IUE included components to describe and access various sensor types, as well as data exchange

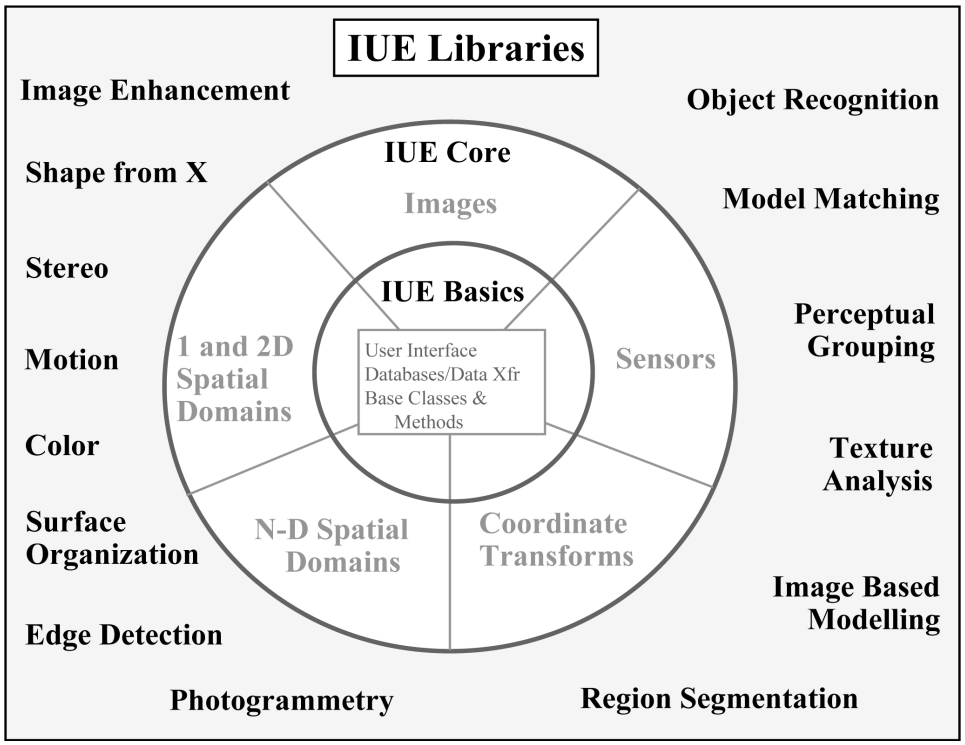

Figure 2.2: Major partitions of the image understanding environment. [46]

components, allowing it to perform the *Access* and *Convert* layers of our scope of computer vision.

The IUE class hierarchy provided a broad coverage of IU constructs, allowing multiple representations in order to offer flexibility to the user. The framework was also designed to be extensible and customizable in order to allow researchers to integrate their algorithms and data structures. Major branches of the class hierarchy include: Base Classes, Spatial Objects, Coordinate Systems and Transforms, Images, Image Features, and finally Scenes and Sensors. Figure 2.3 shows an example of the image feature classes representable using the IUE.

While the IUE functionally covered many of the problems within computer vision which we are interested in, it remains a library focused on providing functionality and does not provide the abstraction over vision algorithms that is the key to our approach. The IUE failed to capture the interest of the computer vision community, and ceased to be supported following 1996.

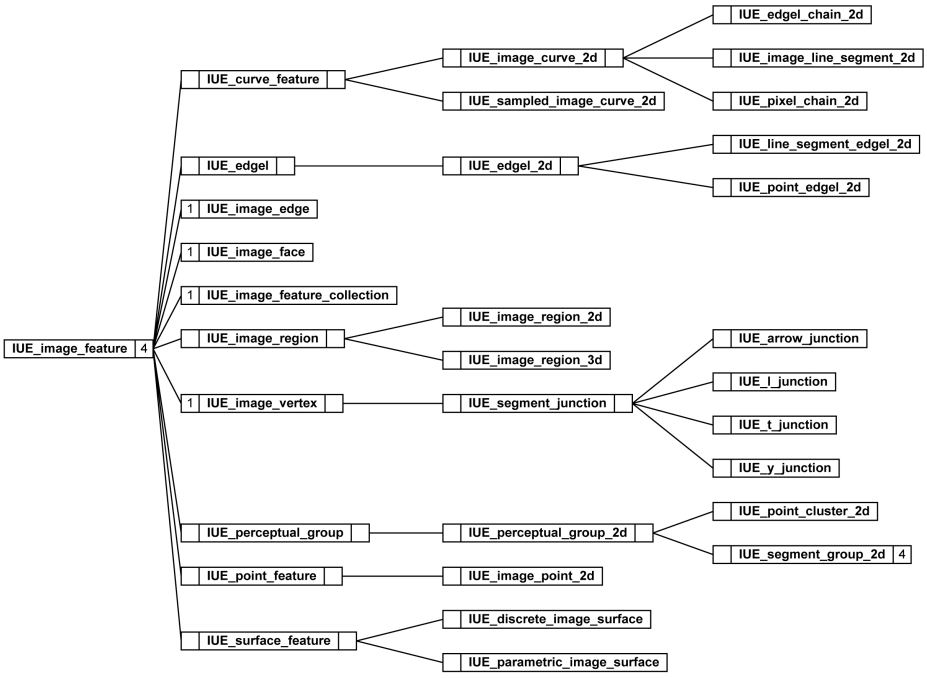

Figure 2.3: Example classes for image features within the image understanding environment. [46]

SIGMA

SIGMA [59] is a framework and control structure of an image understanding system designed for aerial image processing. SIGMA consists of three experts: a Geometric Reasoning Expert (GRE) for spatial reasoning. A Model Selection Expert (MSE) for appearance model selection, and a Low Level Vision Expert (LIVE) for knowledge based picture processing. Figure 2.4 shows the framework of the system.

The focus of SIGMA is within quite a narrow application domain, however each of the component experts provide an example of automation of computer vision techniques, and particularly how these can be combined to create useful functionality.

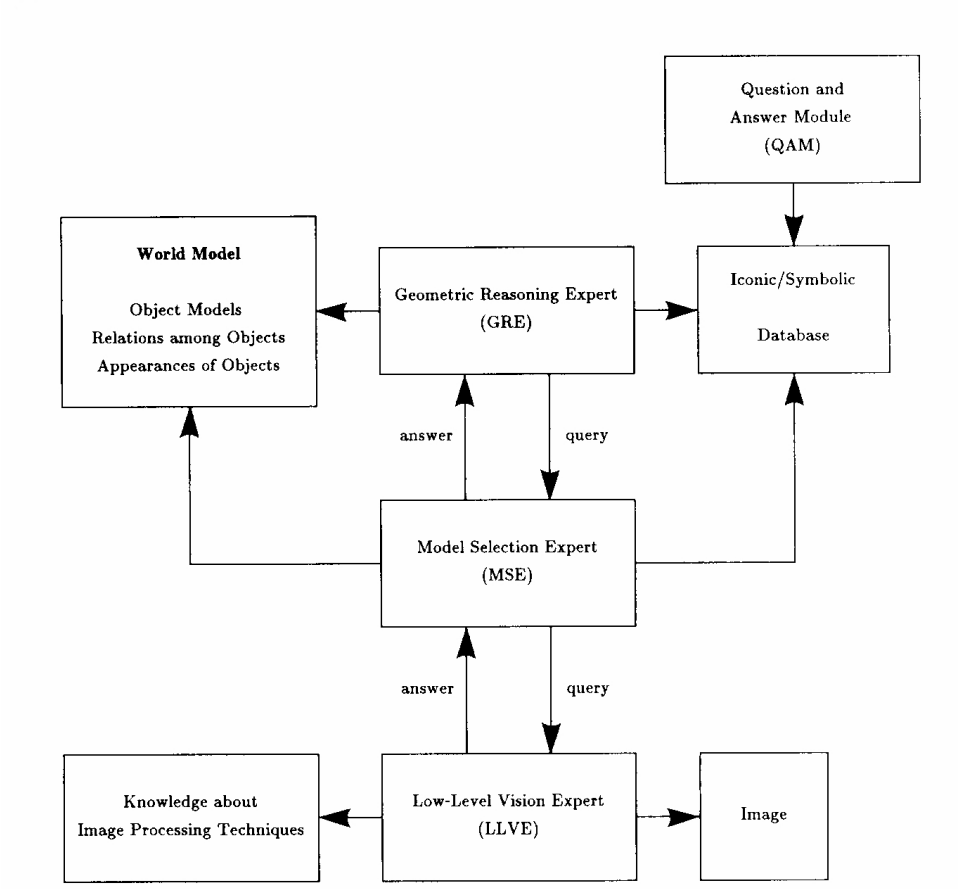

Figure 2.4: The Framework of the SIGMA image understanding environment designed for aerial image processing. [59]

BORG

Borg [20] is a knowledge-based system for the supervision of a library of image processing operators used mainly in biomedical applications. It was designed following the prescriptions of a study on knowledge-based system for operator supervision in image processing. Borg can perform several image-processing tasks in widely varying application domains. Its role is to build a chain of operators that can process every image of a given class. The notion of image class is here restricted to a series of images of the same scene, taken with the same camera and in the same acquisition conditions. In particular, all images have the same resolution. To ensure that the same

chain of operators can process every images of the class, such a chain of operators must integrate control structures to adjust its behavior.

Designing an application involves the active participation of two types of experts: an expert from the domain of application (biology, geography...) to set the problem, and an expert in image processing to formulate the corresponding request by selecting relevant tasks and judicious quantitative and qualitative image features. This problem/request translation is not straightforward; several comings-and-goings are generally needed to get a satisfactory request. In this respect, the system can be seen as an experimentation tool because it allows rapid production of results and favors dialogue and cooperation between both protagonists. The design of an application with Borg follows a three-stage process:

1. Initialization This first step aims at getting a realistic vision of the difficulties of the tasks to be performed and at identifying relevant image features. From a summary description of the application, given by the domain expert, the image-processing expert can suggest a first version of the request. Accordingly, a first version of the chain of operators can then be built by the system and first output images produced;

2. Improvement Through visual assessment of these results, the domain expert is now in a position to refine the request. As a matter of fact, images are the best communication medium between both experts. The request can then be progressively refined, by analyzing various results corresponding to slight variations on its formulation;

3. Validation Once results are judged satisfactory for several images, the validation step consists in testing this same request on the whole set of images representative of the application, to make sure of its robustness. If necessary, the request may be further refined, to take into account particular cases not encountered before.

At the end of this third step, the hope is that the chain of operators actually represents the application under study. The generator of C++ code then produces a program corresponding to this application. Borg offers user-friendly interfaces allowing experts to add new knowledge and new operators in the system as well as the visualization of all potential trees that it can produce to help expert to maintain the knowledge base and the library of operators. Figure 2.5 shows the plan for a BORG cytological application. Clearly significant expert knowledge and collaboration of experts across multiple fields is required to build an application.

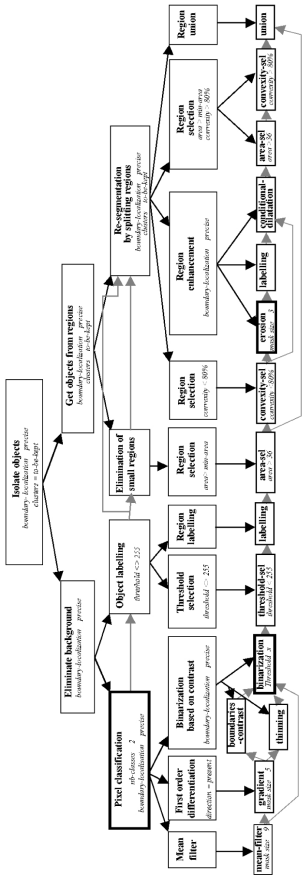

Figure 2.5: The plan for a BORG cytological application. Image data flow at the functionality level and at the operator level are by drawn with gray arrows. [20]

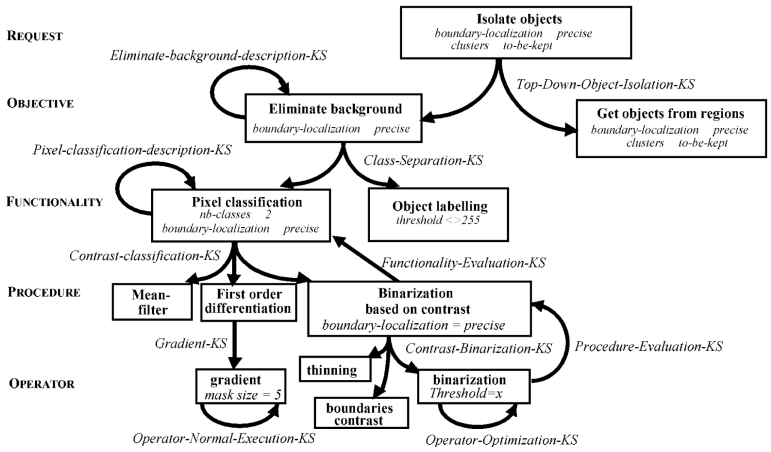

Figure 2.6: Knowledge sources used in the cytology application. [20]

Each small action contributing to the construction of a plan is performed by applications of successive knowledge sources. In the Borg knowledge base, there are five categories of knowledge sources: planning, instantiation, execution, description, and evaluation. With these categories of knowledge sources, correcting the plan is done by proposing new decompositions for tasks that failed, and assessing results is done in two steps: a step when evaluations rules for each task are defined by description, and a step when results are built and evaluated by evaluation. Figure 2.6 shows the knowledge sources used in the cytology application.

2.2.2 Computer Vision Libraries

There have been many attempts at creating a computer vision library that meets the needs of the vision community. The libraries examined below were developed with vision experts in mind, and require significant knowledge of which components to use and how components can be combined to create vision systems. The functionality that they provide is significant, and each has contributed in major ways to the field of computer vision. By examining this functionality, and the method by which it is organized and interfaced, we hope to provide a similar set of image processing and computer vision techniques through a context aware, problem-centric language model.

The IPI includes functionality to do: ,	while specifically excluding:
primitive image manipulation	computer graphics
image enhancement	pattern recognition
image restoration	image understanding
image analysis	multimedia
image classification (basic)	communications
image visualisation (basic)	specific implementations
standard colour models	window systems
image transport sensor	image acquisition
image compression and decompression	image presentation
	device control

Table 2.1: Scope of the Image Processing and Interchange standard.

Image Processing and Interchange

Image Processing and Interchange (IPI) provides a programming interface and an image transfer specification that may be used for a variety of image processing applications; it is not intended specifically for vision applications. It was published as an international standard in 1995. IPI comprises three parts. The first part defines IPI's main architectural features. The second part is concerned with processing images; and the third part considers image interchange. The standard defines a 'functional specification,' a detailed description of the image processing functions that must be provided without reference to any programming language; a separate standard describes how the functional specification is mapped onto a specific programming language.

Images that may be processed by the Programmer's Imaging Kernel System (PIKS), the image processing part of IPI, may be thought of as comprising five dimensions: the usual two spatial dimensions; depth (for representing voxel data); time (for moving imagery); and band (for colour or multispectral data). Multi-sensor data can also be represented by means of the band dimension. Several pixel representations are supported, including unsigned byte, integer, floating-point, and complex. The range of image processing functions is mainly limited to image processing application. Definitions are such that hardware can be built to implement many of the operations. Table 2.1 outlines the scope of the IPI standard. Comparing it with our proposed scoping model for computer vision it seems very well thought out as far as modularity.

Processing operators within IPI perform one of three possible conver-

sions:

- image to image
- image to non-image
- non-image to non-image Converting non-image data to images is the domain of computer graphics and is excluded from IPI.

VISTA

Vista [81] is a software environment supporting the modular implementation and execution of computer vision algorithms developed by Lowe et al. around 1994 at the University of British Columbias Laboratory for Computational Intelligence. It was designed to be extensible, portable, and freely available, and was proposed as an appropriate medium for the exchange of standard implementations of algorithms. Unlike systems that are designed principally to support image processing, Vista provides for the easy creation and use of arbitrary data types, such as are needed for many areas of computer vision research. It uses a flexible, self describing file format so that objects of many types, including custom ones, can be represented for storage and interchange.

Operations were provided both as a stand-alone UNIX program and as library routine. The functionality of Vista included:

- Viewing data: interactive programs for viewing images and edge vectors; toolkit for generating such programs; widget for displaying images with overlaid edges and other graphics; routines for allocating a standard palette of display colors; routine for popping up a window displaying an image.
- Basic image processing: scale, crop, flip, transpose or rotate an image; adjust image brightness and contrast; perform an arithmetic or logical operation on each pixel of an image; convert an image from one pixel representation to another; convert a color image to gray-scale; compute pixel statistics; generate test images; corrupt an image with Gaussian noise.
- Image filtering: convolve an image with a 2D or 3D kernel; convolve an image with a separable filter, such as a Gaussian or its derivative.

- Fourier analysis: build a complex image from real and imaginary components; compute the. Fourier transform of an image; compute the magnitude or phase of a complex image.
- Edge detection: estimate image gradient in 2D or 3D; decompose gradient into magnitude and orientation; mark zero crossings; Cannys edge detector.
- Edge organization: link edge pixels into curves; decompose curves into straight line segments.
- Data file manipulation: combine multiple files into a single, multi-object file; combine multiple images into a single, multi-band image; select specified objects from a file; select specified bands from an image.
- Format conversion: convert images to/from Portable Gray Map (PGM) format; read image from a text file.
- Printing: render images, edge vectors, and flow fields as Encapsulated PostScript (EPS) documents; program for arranging multiple EPS documents on a page; routines for generating PostScript documents.
- Other: estimate optical flow; camera calibration.

OpenCV

OpenCV is a computer vision library originally developed by Intel. It is free for commercial and research use under the open source BSD license. The library is cross-platform, and runs on Windows, Mac OS X, Linux, PSP, VCRT (Real-Time OS on Smart camera) and other embedded devices. It focuses mainly on real-time image processing, using Intel's Integrated Performance Primitives if available to accelerate itself.

Officially launched in 1999, the OpenCV project was initially an Intel Research initiative to advance CPU-intensive applications, part of a series of projects including real-time ray tracing and 3D display walls. The main contributors to the project included Intels Performance Library Team, as well as a number of optimization experts in Intel Russia. In the early days of OpenCV, the goals of the project were described as:

- Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.

OpenCV's application areas include:
2D and 3D feature toolkits
Ego-motion
Gesture Recognition
Human-Computer Interface (HCI)
Mobile robotics
Motion Understanding
Object Identification
Segmentation and Recognition
Stereopsis Stereo vision: depth perception from 2 cameras
Structure from motion (SFM)
Motion Tracking

Table 2.2: Application areas of OpenCV.

- Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
- Advance vision-based commercial applications by making portable, performance-optimized code available for free with a license that did not require commercial applications to be open or free themselves.

It has since grown to include over 500 algorithms centred around computer vision. Table 2.2 outlines OpenCV's main application areas, while Table 2.3 shows some of the additional functionality beyond image processing that is included in order to support these areas.

With functions available to access cameras, convert into different formats, modify the images, and some limited capability for image understanding, OpenCV[12] is commonly used in both academia and industry. Useful function calls such as CornerHarris, CalcHist, Filter2D, Sobel, etc. are provided, however these require considerable knowledge about how to solve a particular image processing problem and offer little insight into what the problem is, making it difficult to adjust when the problem changes. We see our general model of vision as an abstraction that builds on top of algorithmic approaches such as OpenCV to leverage the specialized knowledge contained in their impressive scale of algorithms. Thus, implementations of our model could use OpenCV effectively and allow non-specialists to take advantage of it easily.

Machine Learning components of OpenCV
Boosting
Decision Trees
Expectation Maximization
k-nearest neighbor algorithm
Naive Bayes classifier
Artificial neural networks
Random forest
Support Vector Machine

Table 2.3: To support some of the above areas, OpenCV also includes a statistical machine learning library that contains the functionality listed above.

As can be seen in Table 2.2 OpenCV does not separate out the different elements into thin layers of abstractions. Based on Marenko et al.'s work [56], this also presents challenges for reuse and scalability in real-world contexts, as has been encountered numerous times by the author.

2.2.3 Medical Image Processing Libraries

Medical image processing is an important subset of the vision community, which has developed a number of image processing libraries specific to the medical field. These libraries are specifically designed to deal with the multi-sensor, often 3-dimensional data that is common in the field. Although OpenVL does not initially intend to support this functionality, many algorithms are common to both computer vision and medical image processing, and it is a logical path of future development. Additionally, a significant proportion of the medical imaging community are medical experts and not vision experts, providing us with insight as to how to provide functionality in a way that is useful to non-vision specialists.

NA-MIC

The National Alliance for Medical Imaging Computing (NA-MIC) is a multi-institutional, interdisciplinary team of computer scientists, software engineers, and medical investigators who develop computational tools for the analysis and visualization of medical image data. The purpose of the center is to provide the infrastructure and environment for the development

of computational algorithms and open source technologies, and then oversee the training and dissemination of these tools to the medical research community.

The NA-MIC Kit [69] is a free open source software platform. The NA-MIC Kit is distributed under a BSD-style license without restrictions or give-back requirements and is intended for research, but there are no restrictions on other uses. It consists of the 3D Slicer application software, a number of tools and toolkits such as VTK and ITK, and a software engineering methodology that enables multiplatform implementations. It also draws on other best practices from the community to support automatic testing for quality assurance. The NA-MIC kit uses a modular approach, where the individual components can be used by themselves or together. The NA-MIC kit is fully-compatible with local installation (behind institutional firewalls) and installation as an internet service. Significant effort has been invested to ensure compatibility with standard file formats and interoperability with a large number of external applications.

Users of the NA-MIC Kit will typically use a combination of its many modular components. 3D Slicer is a general purpose application. Biomedical researchers use this software tool to load, view, analyze, process and save image data. Slicer has been implemented to interoperate with many other tools, including XNAT, which is an open source image database. Slicer modules, which are dynamically loaded by Slicer at run-time, can be used to extend Slicer's core functionality including defining graphical user interfaces. Modules are typically used by algorithms and application developers. Figure 2.7 shows the slicer gui.

Application and algorithms developers may also use NA-MIC Kit toolkits and libraries. The Insight Segmentation and Registration Toolkit ITK can be used to develop slicer modules for medical image analysis. The Visualization Toolkit can be used to process, visualize and graphically interact with data. KWWidgets is a 2D graphical user interface toolset that can be used to build applications. Teem is a library of general purpose command-line tools that are useful for processing data.

The ITK registration toolkit [68] is of particular interest as it provides image registration functionality to a number of researchers working within the medical imaging community. In Chapter 6 we use several of these algorithms in our testbench, exploring how well each is able to provide a solution to a range of image registration problems. Within ITK registration is modeled as a series of components which combine to create a registration algorithm. Figure 2.8 introduces this model. A *metric* measures the error in alignment, an *optimizer* iterates, solving for a *transform* which contains the

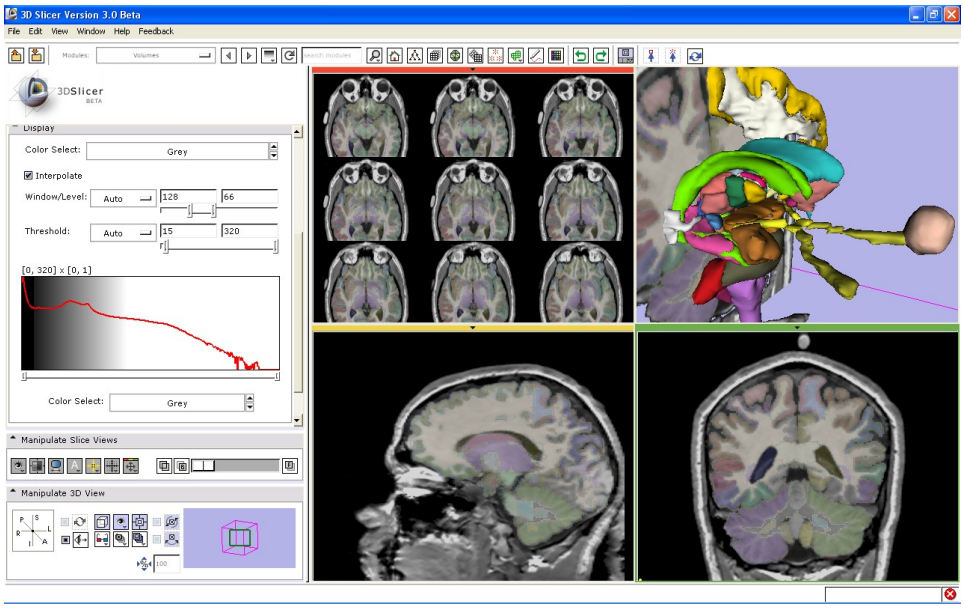

Figure 2.7: The NA-MIC 3-D Slicer tool gui. [69]

minimum error between the two images. This transform is applied to the moving image using an *interpolator*, which interpolates the moving image within the fixed image's space. The process iterates until a solution is found or certain criteria are met.

The components of this process are designed to be interchangable, theoretically allowing a developer to select a different image type, metric, optimizer, transform, and interpolator as is appropriate for their problem. In practice this turned out to be much more difficult as many components are incompatible or unimplemented in the current version of ITK, necessitating major changes to the source code whenever one of these was changed. Further obfuscating the process was the lack of documentation and meaningful error reporting in instances where components were incompatible.

2.2.4 Visual Programming Environments for Vision

Visual programming for computer vision was developed as an attempt to simplify development, making it both faster and more accessible to non-experts. These environments allow developers to piece together computer vision or more commonly image processing components in a visual flow chart, rather than through code, connecting the outputs of one module to the in-

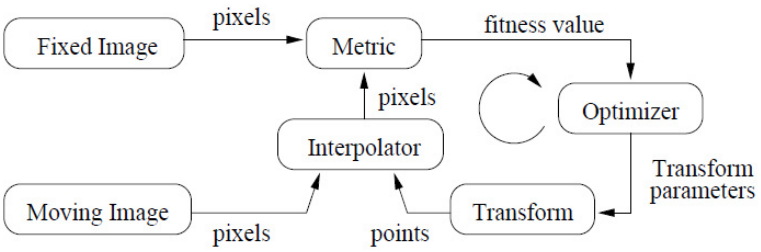

Figure 2.8: The ITK Model of Image Registration. [68]

puts of the next to create an application. Knowledge of the components and how they must be connected in order to achieve the desired application, particularly when dealing with data types other than images, means that developers must still understand a considerable amount of expert vision knowledge. To counteract this requirement image processing functions dominate these environments, with the majority of components taking an image as input and similarly outputting an image, making them more suitable to the *Modify* layer of the CV scope, however insight can be gained by examining the functionality and methods within.

Khoros

Khoros [47] is an integrated software development environment for information processing and visualization on SUN machines, developed by Konstantinides et al. at the University of New Mexico in the late eighties and early nineties. Khoros components include a visual programming language, code generators for extending the visual language and adding new application packages to the system, an interactive user interface editor, interactive image display programs, surface visualization, an extensive library of image processing, numerical analysis and signal processing routines, and 2D/3D plotting packages. The data processing libraries contain over 260 programs, in the following categories: arithmetic, classification, color conversion, data conversion, file format conversion, feature extraction, frequency filtering, spatial filtering, morphology filtering, geometric manipulation, histogram manipulation, statistics, signal generation, linear operations, segmentation, spectral estimation, subregion, and transforms.

The Khoros system provides a high-level, abstract visual programming language, Cantata that allows the researcher to control the data flow of

Arithmetic	
Unary Arithmetic	Scale, Normalize, Invert, Clip, etc.
Binary arithmetic	Add, Subtract, Multi ply. , Blend, etc.
Logical Operations	And, Or, Xor, Shift.
Image Processing	
Spatial Filter	Sobel, Median, 2-D convolution, Edge Extraction, etc.
Morphology	Erosion, Dilation, Skeletonization, etc.
Transforms	FFT, Hadamard.
Frequency Filters	LPF, BPF, HPF, Band-Reject, Inverse, Wiener Restoration, etc.
Geometric Manipulation	Shrink, Rotate, Transpose, Interactive image warping, etc.
Subregion	Extract, Insert, Pad, etc.
Image Analysis	
Segmentation	Threshold, Medial Axis Transf., etc.
Feature Extraction	Shape analysis, Region Matching, Fractal Analysis, Texture Extraction, etc
Classification	K-means, Labeling, LRF-classifier, etc.

Table 2.4: Khoros Routines for image processing.

images. Central to the Khoros system is the need for a consistent yet flexible user interface development system that provides cohesiveness to the vast number of programs that make up the Khoros system. Automated tools were developed to assist in maintenance as well as development of programs. The software structure that embodies this system provides for extensibility and portability, and allows for easy tailoring to target specific application domains and processing environments. Table 2.4 outlines the image processing functionality provided by the Khoros system.

Cantata is the visual programming environment built within the Khoros system. Cantata contains many features not typically found in visual programming environments such as visual hierarchy, iteration, control structures, expression-based parameters and program encapsulation. In Cantata, visual programs are created as directed graphs, where each node of the graph is an iconic element representing a program and each directed arc represents a path over which data flows. By connecting the data paths between programs user can interactively draw out a solution in a more natural way that matches their mental representation of the problem. By providing a visual environment for problem solving, their claim is that Cantata increases the productivity of both researchers and application developers, regardless of their programming experience. Although the visual programming language provides an easier mechanism for quick development of image processing applications from existing algorithms, we would argue that developers still require significant expert knowledge of which algorithms to use when, and how to combine components.

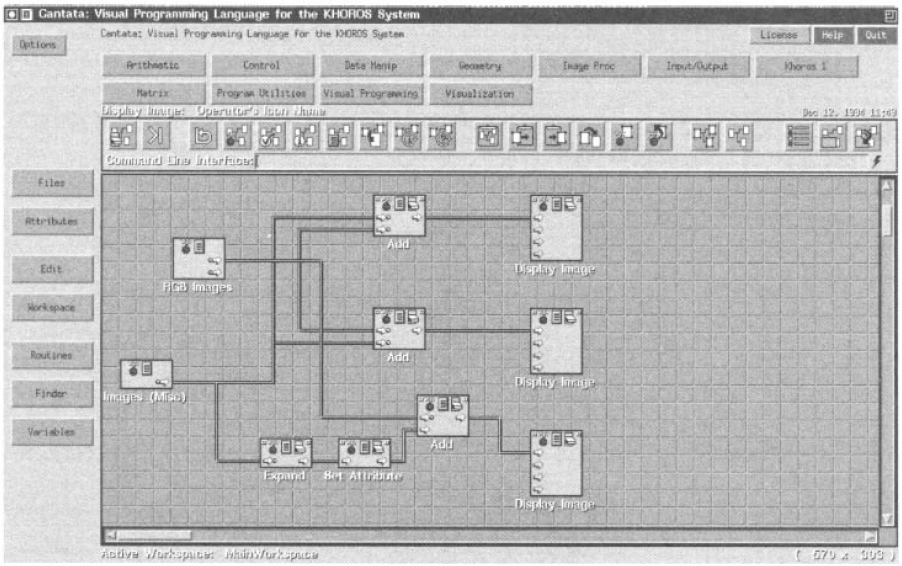

Figure 2.9: The Cantata GUI running a vision processing program using the Khoros System. [47]

Icons called glyphs represent programs from the Khoros system. Each of the hundreds of stand-alone data processing and scientific visualization programs in the Khoros system can be represented in Cantata as glyphs. When accessed in the visual language, a Khoros program is referred to as an operator. To create a visual program, the user selects the desired programs, places the corresponding glyphs on the workspace, and connects the glyphs to indicate the flow of data from program to program, forming a network within a workspace. Control structures can be used to branch and merge data flow, or to implement loops. Workspaces can be executed, saved and restored to be used again or modified later. Workspaces may also be encapsulated into independent applications with a very simplifled graphical user interface so that they may be treated as stand-alone Khoros applications. Figure 2.9 shows the Cantata GUI running a vision processing program.

Visual hierarchy, iteration, flow control, and expression-based parameters extend the data flow paradigm to make Cantata a powerful simulation and prototyping system. Data and control-dependent program flow is provided by control structure glyphs such as if/else, while, count, and trigger. Visual subroutines, or procedures are available to support the development of hierarchical data flow graphs. Variables may be set interactively by the

user, or calculated at run time via mathematical expressions tied to data values or control variables within the visual network.

Apple CoreImage and CoreVideo

Apple provides an abstracted image processing pipeline using its collection of components including CoreImage and CoreVideo combined in Quicktime7. These components support access, conversion and modification operations of image data coupled to a rendering framework. CoreImage abstracts pixel-level manipulations into stackable plugins, allowing applications to take advantage of acceleration of image processing without requiring the addition of hardware-specific code. Each image unit specifies a filter, transform, or other effect which can be applied to the original pixel data without modifying the original data. CoreVideo provides mechanisms to convert media data coming from image sources into representations suitable for rendering in a media player. These two components are accessible through Apple's Quartz Composer [4], which provides a visual programming environment which simplifies the development of simple Vision applications. While this framework is effective for viewing and modifying media content, it lacks an abstraction for the analysis component of computer vision, which OpenVL attempts to fill.

Mac OS X 10.4 includes 100 standard Image Units, including the following:

- Median, Gaussian, Motion and Zoom blurs
- Noise Reduction
- Color adjustment: Exposure, gamma, hue, and white point
- Distortions: Pinch, hole, displacement, glass, torus Lens, twirl, vortex, circle splash and circular warp
- Generators: Star shine, sunbeams, checkerboard and lenticular halo
- Color blends: burn, darken, difference, exclusion, hard light, hue, lighten, luminosity, multiply, overlay, saturation, screen, soft light
- Geometry: Crop, scale, rotation, Affine transformation
- Halftone, tile, and style filters
- Transitions: Swipe, flash, page curl, copy machine, disintegrate, and dissolve

2.2.5 Declarative Languages for Image Processing

One important aspect of OpenVL is that it provides a language model to programmers to describe *what* they want done rather than *how* to do it. In this way, it behaves somewhat like a declarative approach to problem solving. Examining previous attempts at declarative image processing provides us with significant insight into the possibilities and challenges of a language model of this nature.

ShapeLogic

ShapeLogic[88] is an open source toolkit for declarative programming, image processing and computer vision. It has two main applications: The Color Particle Analyzer will find and categorize particles on a relatively uniform background then make a report of geometric properties for each of the particles. The main application is for recognizing cells in medical image processing. The Letter Matcher is a general categorizer for skeletonized lines.

While ShapeLogic's declarative programming system was initially developed for image processing and computer vision, it is widely applicable. ShapeLogic is intended as basic plumbing software that turns a logic engine or a neural network into a simple plugin component to ease entry into vision and image analysis. ShapeLogic also fills gaps missing from current Java image processing libraries.

ShapeLogic takes a more literal approach and uses a logic language to implement their declarative constructs for vision processing. OpenVL specifies a language model and state machine but does not require implementation in any particular language or an inference engine to do constraint satisfaction making it simpler to implement and accelerate.

FVision

FVision is a functional reactive programming (FRP) library written in Haskell that targets visual tracking. Low level vision processing functionality is provided through links to XVision C++ libraries, allowing the system to maintain 90% of the speed of the purely C++ implementation. The resulting system combines the overall efficiency of C++ with the software engineering advantages of functional languages: flexibility, composability, modularity, abstraction, and safety. Figure 2.10 shows the feedback loop of a primitive tracker built in XVision.

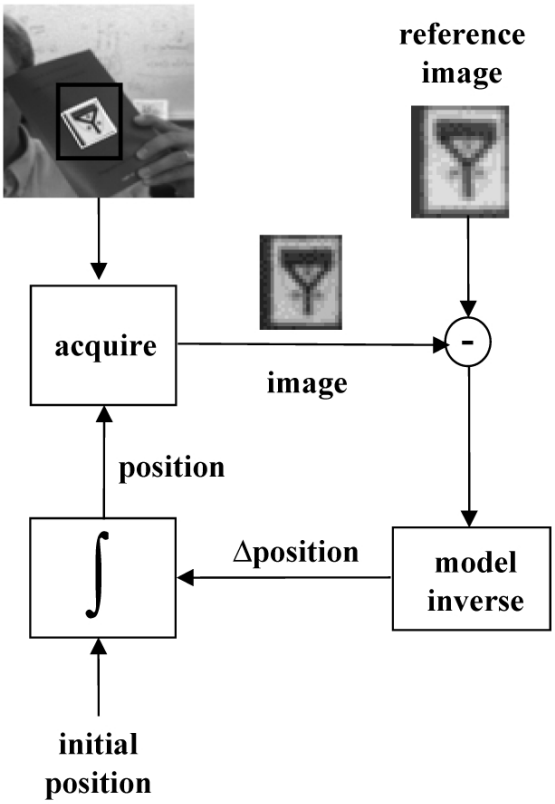

Figure 2.10: Feedback loop of a primitive tracker built in XVision.

A similar primitive FVision tracker, the SSD tracker, has been developed that is fully declarative. Given a reference image, the observer pulls in a similar sized image from the video source at the current location. The stepper then compares the image from the current frame with the reference, returning a new location and a residual. This particular tracker uses a very simple location: a 2-D point and an orientation. The SSD observer is an XVision primitive. Figure 2.11 shows source code for a composite tracker built in FVision.

```
faceTrack :: Image ->
  Tracker (EitherT (EitherT SizeAndPlace SizedAndOriented) Residual)
          (Either (Either () Point2) Transform2)
faceTrack image =
  tower (tower motionDetect blob (upFromMD, downFromBlob))
        ssd onlyRight (upFromBlob, downFromSSD)
 where
   upFromMD mt =
      if mArea mt > mdThreshold then Just mCenter mt else Nothing
   downFromBlob mt =
       if blobSize mt < bThreshold then Just (blobCenter mt) else Nothing
   upFromBlob mt =
      Just (translate2 (blobCenter mt)
              `compose2`
             rotate2 (blobOrientation mt))
   downFromSSD mt =
     if residual mt > ssdthreshold
         then Just (origin2 `transform2` (valueOf mt))
         else Nothing
```

Figure 2.11: A composite tracker built in FVision.

2.3 Summary

This chapter provides a literature review of both image registration techniques and of computer vision libraries and frameworks, attempting to cover both of these topics in sufficient detail that readers gain an understanding of the research that this thesis is built upon. While this is not an exhaustive survey, the libraries and frameworks discussed in this chapter represent a broad sampling of prior work and existing computer vision tools available to developers.

Chapter 3

A New Taxonomy of Image Registration

"I had a lot of trouble with engineers, because their whole background is learning from a functional point of view, and then learning how to perform that function."

- Brian Eno

Existing taxonomies of image registration [14, 93, 103] summarized in Chapter 2 divide the field algorithmically, focussing on *how* registration is accomplished. This approach provides a good basis for classifying and comparing algorithmic similarity, however it does not focus on the conditions important to the problem of registration. Although the field is rapidly moving towards automatic image registration, algorithms are often targeted towards a single application area such as stitching panoramas, super-resolution, high-dynamic-range imaging, focal stacking, multimodal imaging, video summarization and stabilization, or satellite image analysis.

The registration methods required for these applications can also be used on a limited subset of problems from the other domains. Understanding where algorithms perform well and where they fail is crucial when designing a system which requires registration. Under the previous framework and taxonomy this knowledge was largely empirical, with each algorithm roughly mapped to a single corresponding application whose problem conditions are (again empirically) well known, although often not well defined. Knowledge of which algorithm performs best from amongst those that target the conditions of a particular application area is similarly empirical and is often contentious as the conditions or images used to represent those conditions are often not the same. Binary classification of a problem type does not allow for the level of distinction required to know how effective a given

algorithm will be within a particular range of the problem space. This information also changes regularly as new algorithms are developed. Without a model of the image registration problem space, expressing the conditions of a given problem is not a well defined process.

The taxonomy proposed within this chapter maps the registration problem space based directly on the conditions of the registration problem being solved. The most appropriate method of registration of a set of images can be determined by examining how the properties of the images or data being registered vary in comparison to one another. By taxonomizing the problem in this way we change the abstraction of registration from one requiring knowledge and expertise about particular algorithms, both in how they work and when to use them, to one requiring expertise about the conditions surrounding the registration problem itself, which is much simpler for programmers who are not vision experts to understand and specify. We provide a binary mapping of the papers and techniques explored in Chapter 2 into our n-dimensional space of problem variances, based on their reported problem area and performance within the literature. Even though they are typically designed to solve problems along a single dimension of variation, individual algorithms can be thought of as supporting a volume of solutions within the problem domain. The dimensions of this volume are difficult to determine, however, without doing a direct evaluation of the algorithm under different conditions.

Within our framework image pairs are organized according to the categories: non varying, intensity varying, focus varying, sensor varying, and structure varying. In practice, examining the types of variations that occur in a pair or sequence of images allows photographers to select an appropriate application or researchers to select an appropriate algorithm in order to find the best alignment. Basing our taxonomy around this understanding of the image registration problem space makes the field more accessible to non-experts, and as we will see in Chapter 7 allows for further automation of image registration.

An examination of the literature provides some guidance as to which problem types a particular algorithm should be good at, but reporting is often based on a visual analysis of the alignment of a small number of image pairs chosen by the authors. The problem is even greater for non-experts, who must navigate an algorithm centric model and rely on these reports when selecting an algorithm. Image registration methods are often presented directly in relation to particular applications which require this functionality. Examining applications that rely on image registration from a data centric perspective provides insight into the relative success of indi-

vidual algorithms when particular conditions occur. Our main dimensions of variation are examined below in detail and algorithms that are invariant across each dimension are presented. The result is a taxonomy in which algorithms do not occur in a single place, but rather occur in multiple categories. In fact, categories are crudely representative of the n-dimensional problem space that registration algorithms are attempting to cover, while specific algorithms support a volume of solutions within the n-dimensional problem space. In Chapter 5 we examine the development of a testbench which allows for a more sophisticated evaluation of algorithm performance under particular conditions. As we will see in Chapter 6, by identifying the position of registration problems within this n-dimensional problem space it is possible to select one or more algorithms that were appropriate on other problems with similar conditions.

Brown's framework touches on how knowledge of the types of variation that occur in image sets can be used to guide selection of the most suitable components for a specific problem. These variations are divided into three classes: variations due to differences in acquisition that cause the images to be misaligned, variations due to differences in acquisition that cannot be easily modeled (such as lighting or camera extrinsics), and finally variations due to movement of objects within the scene. These are labeled by Brown "corrected distortions", "uncorrected distortions", and "variations of interest" respectively. Zitová and Flusser provide a model of variation according to the manner in which the images were acquired: different viewpoints, times, conditions, sensors, and finally scene to model registration. Within their survey they do not use this mapping directly, however in many cases they discuss the type of problem each method has been designed to solve, allowing a similar mapping of methodology from situation. Similarly Pluim et al.'s "aspects of the application" entail image modalities, subject of the registration, and the object of registration. This delineation provides an excellent starting point for variations that are important within the medical imaging community. It is this concept of variations that we have chosen to base our taxonomy on, extending these initial ideas into specific variations common in image registration and exploring the successful algorithms under different problem conditions.

Image registration problems are mapped below into the following categories: non varying, intensity varying, focus varying, sensor varying, and structure varying. Each categorization represents a dimension of the image registration problem space.

Figure 3.1: Input images which contain only spatial variation, and the corresponding rendered image derived by stitching the set using a feature based method.

3.1 Only Spatial Variations

The problem of image registration is one of finding the spatial variation between a pair of images. Image pairs that differ purely spatially are the most common type of image registration problem. Figure 3.1 shows a number of example image pairs that feature spatial variation. Applications that require registration of images that vary spatially include panorama stitching, super resolution, and remote sensing. Area based methods derivative of Lucas and Kanade [53] are capable of solving these types of registration problems, however feature based methods are the most common technique applied and are generally faster and considered much more accurate unless the image pairs contain little high-frequency information from which to find and match features.

Brown et al [61] represents a sophisticated approach to feature based image registration, using SIFT [52] features to align the images. Other feature descriptors have been attempted including shape context [8], steerable filters [30], PCA-SIFT [43], differential invariants [45], spin images [48], SIFT, complex filters [86], and moment invariants [97], however a comparison by

Mikolajczyk and Schmid [63] suggested that SIFT-based descriptors perform best.

3.2 Intensity Variations

Image pairs that contain significant intensity variations include those used for high dynamic range (HDR) imaging, pairs with varying illumination, and even panorama sequences where light sources shine directly into the lens. Figure 3.2 shows two example images and the resulting HDR image created by aligning and combining them.

Figure 3.2: Input images taken from an HDR set, and the corresponding rendered image derived by combining the aligned pair using a simple tone mapping algorithm.

Feature based methods listed in Section 2.1.2 work for small variations in intensity, particularly if their feature descriptors are gradient based. However for image pairs with significant intensity variation, the interest regions

of feature based methods do not occur in the same location. Sand and Teller [85] attempt to handle intensity varying pairs by only selecting features from parts of the image that can be more easily matched while avoiding parts that are difficult. A combination of area based methods (described below in Section 3.3) and feature based methods are used to iteratively align the images, ignoring feature points that occur in over or under exposed regions of the image. Their technique was designed for matching two video sequences, achieving good results with the limited spatial variation that entails, however it was not tested on still photographs. More recently Tomaszewska and Mantiuk [94] presented a similar idea, reporting a high quality alignment such that the "photographs were aligned with sufficient accuracy so that there are no visible artifacts in the final HDR image" by using only features that occur across all images in the set. These methods of culling features that are inappropriate only work when enough features remain to make a proper alignment.

Schechner and Nayer [87] presented an alignment method based on pyramids of maximum likelihood as a part of their approach to generalize panorama images to incorporate variations in intensity. The addition of uncertainty into the intensity based search space allows for a much better alignment under these conditions. Figure 2.1 shows the corresponding mosaic regions of an unaligned, intensity aligned, and maximum likely aligned set of images. Kang et al. [42] also described a technique for creating high dynamic range video from a sequence of alternating variable intensity exposures. Their sophisticated HDR stitching process uses local alignment and motion estimation to compensate for camera movement and object motion within the scene, a technique tailored to their input data.

Finally, Ward [82, 100] introduced a method specifically designed to align images with significant variations in intensity. The technique thresholds image pairs into pyramidal bitmaps, creating binary images that represent regions that are neither over nor underexposed. The bitmaps are analyzed and aligned for translation errors using shift and difference operations at each level of the pyramid. With this method three megapixel image sets are aligned in a fraction of a second. Unfortunately their method deals solely with translation errors, although they discuss the possibility of solving for rotation errors, stating that 10% of their data set failed as a result. This binary 'pass' / 'fail' evaluation of the registrations is indicative of the poor evaluation techniques used by researchers in the field.

3.3 Focus Variations

Focus variations can occur in image pairs deliberately as is the case with focus stacking, or through motion blur due to movement of the camera or objects in the scene as can sometimes be the case in super resolution imaging. Figure 4.9 shows two images from a focal stack, and the corresponding image creating from combining the two to maximize the in focus regions.

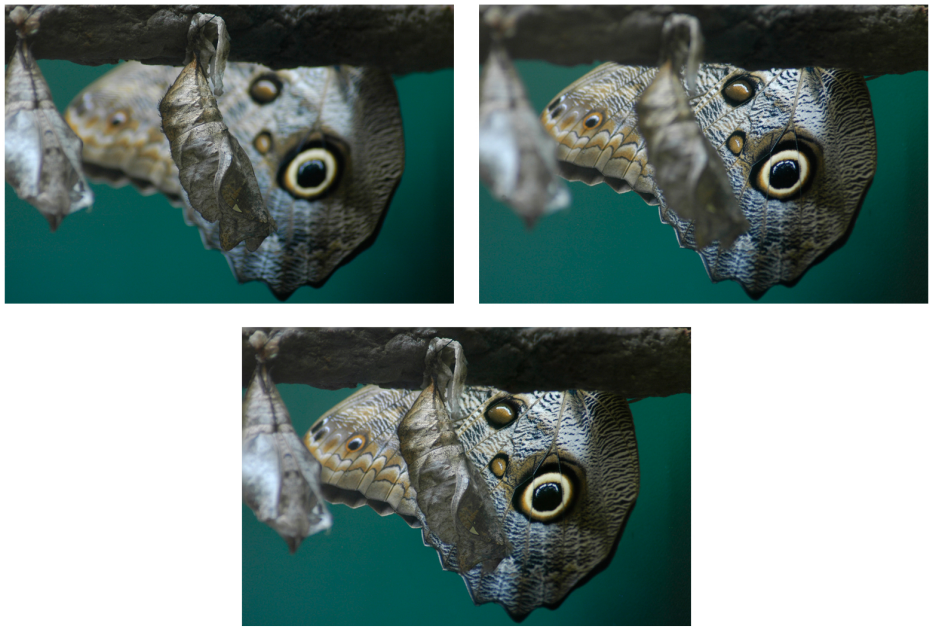

Figure 3.3: two images from a focal stack, and the corresponding image creating from combining the two to maximize the in focus regions.

Here again feature based methods surveyed in Section 3.1 have utility to a point, however image pairs with no overlapping regions of focus do not work; the same interest points are not detected across images with different focal planes. Flusser, Zitová and Suk attempt to overcome this by finding blur invariant interest regions using complex moments [29, 104]. Their presented evaluation consists of a single 128×128 pixel image using the distance of 30 control points candidates from their target positions as their measure of accuracy. This makes it difficult to provide significant conclusion about the quality of registration this provides, although the measured points were at best 3 Euclidean pixels off, and at worst over 8 pixels, leading the author to

believe that it is quite poor.

An alternative commonly used in focal stacking applications are classic intensity based area methods. Lucas and Kanade [53] based sum of square difference area methods are reasonably successful, assuming the spatial overlap between images is significant. This is often the case for focus stacking problems, particularly those composed of microscope data where sensor movement is minimal between images. Bradley et al. [11] make use of normalized cross correlation in their virtual microscopy system requiring an overlap of at least 45% between image pairs and ignoring results that fall outside their expected solution area.

Area based methods work by comparing some measure of the aligned pixel values as an error function to be minimized. Typically for focus stacking a sum of square difference of the pixel intensity is used. The process is an iterative gradient descent: at each iteration calculating the current error, and using the slope of the error space at that point and an estimate of the Jacobian to calculate the next position. This method is susceptible to local minima, but works surprisingly well, particularly when applied at multiple scales via image pyramids.

3.4 Sensor Variations

Image pairs taken using different sensors are commonly referred to as multimodal image pairs. They are common to both medical imaging and to remote sensing applications, where proper alignment of two or more modalities provides significant additional information. When sensors differ there is no guarantee that intensities, gradients, or edges will be similar, and both feature based and intensity based methods fail to find alignments. Figure 3.4 shows an example of a multimodal image pair which contains T1 weighted, T2 weighted and PD weighted MRI scans of the human brain.

Sharma and Paval [89] propose to overcome this difference in intensity by making the images as similar as possible, transforming images into representations invariant to polarity reversals before applying traditional area based techniques. Irani and Anandan [37] similarly transform images into high-pass energy images which are significantly less sensitive to sensor variations. These methods have been further developed by Liu et al. [50, 51] who use Gabor filtering as their local frequency representation. More recently, Henn and Witsch [34] define two nonlinear distance functions and minimize these to find the optimal alignment.

In the field of medical imaging, maximization of mutual information,

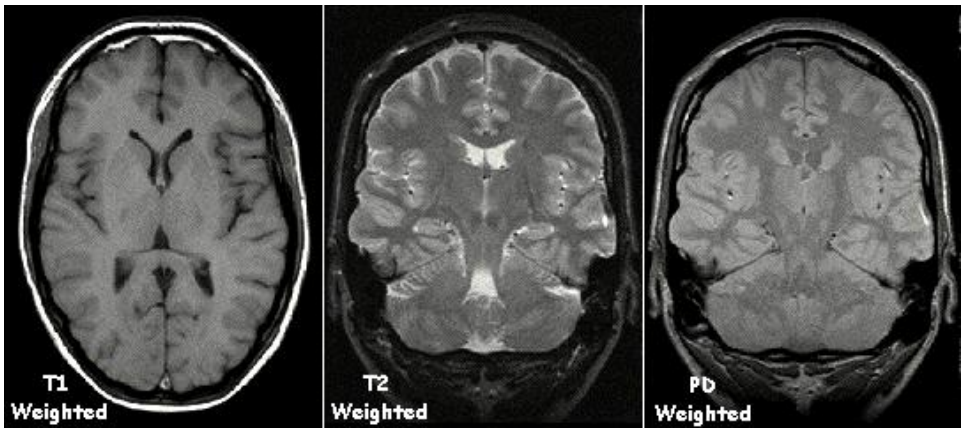

Figure 3.4: Three types of MR image: the T1 weighted image depicts relatively bright grey matter and dark CSF; the T2 weighted image highlights the CSF, while the PD weighted image shows little contrast between tissues. [41]

developed simultaneously by Viola and Wells [98] and by Collignon et al. [54], is the most common method used and was comprehensively surveyed by Pluim, Maintz and Viergever [80]. Bardera, Feixas and Boada proposed two new similarity measures based off of Jensen's difference applied to Rẽnyi and Tsallis-Havrda-Charvãt entropies claiming that their proposed measures are more robust than the normalized mutual information for some modalities and a determined range of the entropy parameter. Gan et al. [31] suggest using Kullback-Leibler distance if a priori knowledge of the joint intensity distribution is available. Makela et al. [57] provide an overview of further methods focusing specifically on cardiac images. While features based on mutual information are being used in pattern recognition [78, 95] they have not to our knowledge been used successfully for registration of multimodal images.

Automatic registration of 3D shapes and volumetric slices of sensor data are common within the medical imaging community, however they fall outside the scope of this thesis.

3.5 Variations in Structure

Image pairs can vary significantly in the structure of the scene they depict, either because objects within the scene have moved, or more commonly

in medical imaging because objects have changed over time. Figure 3.5 demonstrates an the alignment of an MR taken before a surgery and PET scan taken after. Aligning images in spite of these changes often requires non-rigid transforms that solve for the alignment of regions or at the extreme of individual pixels. The selection of points on the grid of solutions, and interpolation of values between those points, along with the algorithms used to solve for the global and local solutions vary from algorithm to algorithm and are the main differentiating factors within this type of registration. Methods discussed here can include a variety of other variations, although sensor variations are most common. This concept of multiple dimensions of variation is discussed further in Section 3.6.

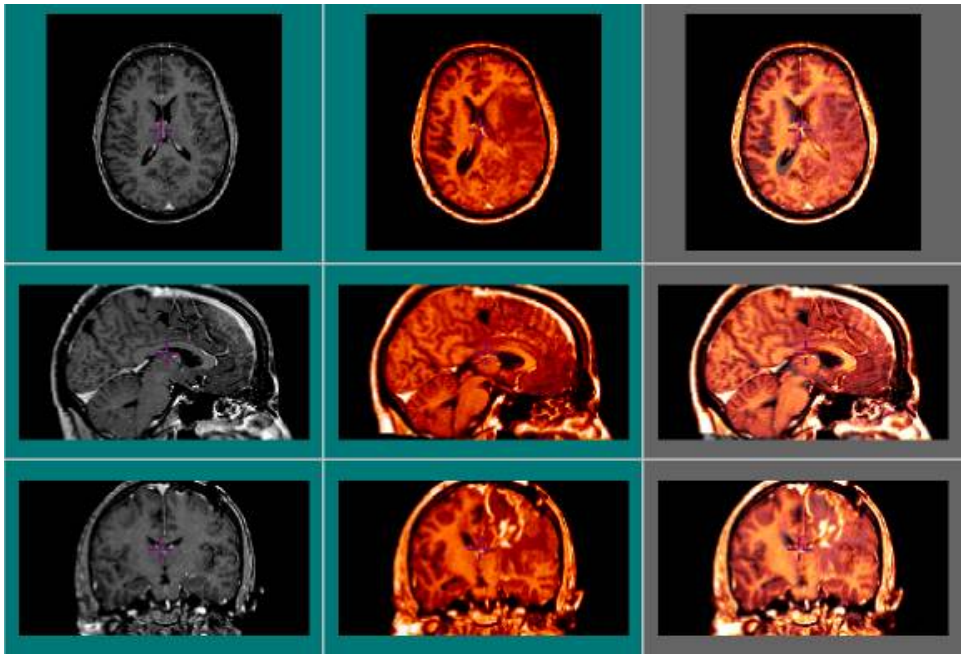

Figure 3.5: Alignment of an MR taken before a surgery and PET scan taken after. [24]

Bookstein uses thin plate splines to interpolate affine transforms [9], reporting significant success, while Moshfeghi's model is similar, but based on elasticity [66]. Christensen et al. utilize a viscous fluid representations of deformable registration [19], and Bro-Nielson and Gramkow [13] have accelerated this concept using a fast fluid model. McInerney and Terzopoulos [62] have surveyed nonrigid techniques and their application within the medical

imaging community.

Rueckert et al. [84] present a nonrigid method that uses a global affine transform, followed by a local B-spline matching of normalized mutual information voxels, applying their technique to the registration of breast MR images. Rohde et al. [83] make several contributions to the field, pioneering use of radially symmetric basis functions rather than B-splines to model the deformation field, a metric to identify regions that are poorly registered and over which the transformation needs to be improved, partitioning of the global registration problem into several smaller ones, and creating a new constraint scheme that allows transformations that are topologically correct. They compare the proposed approach to more traditional ones listed above and show that their new algorithm compares favorably to those in current use. More recently D'Agostino et al. [22] propose modeling the registration as a viscous fluid that deforms under the influence of forces derived from the gradient of the mutual information registration criterion, validating their method by matching simulated T1-T1, T1-T2 and T1-PD MRI images.

3.6 Summary

Our mapping of the image registration problem domain focuses on the types of variation that occur between images to be registered. Conceptualizing these forms of variation as dimensions of the image registration problem, we have an abstraction which allows us to think of algorithms as volumes within the n-dimensional problem space. Existing algorithms have been introduced into this mapping according to the main form of variation that they have been designed to support. Instances where these algorithms have been reported as supporting other forms of variation have been outlined, providing an initial mapping of the space that is summarized below in Table 3.1. The reorganization of image registration into our variation-centric taxonomy provides a basis for several opportunities that advance the field of image registration:

First, using variations as a starting point it is possible to create a model with which to describe registration problems. By describing problems in terms of the variations that exist in their images instead of their algorithmic details it becomes possible to compare image registration problems across different algorithms. With this comparison of which algorithm performs best under a given set of conditions we can create a system that automatically selects the most appropriate algorithm from those available. This model for

image registration is explored in detail in Chapter 4, while the method for interpreting it is outlined in Chapter 6.

In Chapter 7 this concept is extended further into the fully automatic classification of registration problems [72, 73]. Many of the variations explored in this chapter can be automatically detected from the image pairs, allowing for an estimation of where the pairs exist in problem space. This allows for the selection of an algorithm appropriate to the given conditions of the problem. Such an automated system would come close to the 'ultimate registration method' described by Zitová and Flusser at the end of their survey; a system able to recognize the type of task and to decide by itself about the most appropriate solution.

In addition, most image registration methods are designed to work along a single dimension, however the combination of these is becoming more common, particularly in sensor / structure combinations for multimodal non-rigid medical imaging. Another notable multidimensional example is Schechner and Nayer's HDR panorama stitching method [87]. The examination of other combinations of variation such as focus and structure could prove interesting.

Finally, with this taxonomy, and the subsequent model built from it, it becomes possible to develop a testbench for image registration problems that spans a wide range of problem domain. An attempt at this testbench has been undertaken and is introduced in Chapter 5.

Type of Variation	Invariant	Capable	Barely Capable
Purely Spatial	Belongie02[8], Brown03[61], Freeman91 [30], Ke04 [43], Koenderink87 [45], Lazebnik03 [48], Lowe04 [52], Maneewongvatana01 [58], Schaffalitzky02 [86], Gool96 [97], Yang07 [102]	Baker04 [6], Gan04 [31], Henn03 [34], Lucas81 [53], Maes97 [54], Viola97 [98]	
Intensity	Kang03 [42], Reinhard05 [82], Sand04 [85], Schechner03 [87], Tomaszewska07 [94], Ward03 [100]	Brown03 [61], Gan04 [31], Henn03 [34], Lowe04 [52], Maes97 [54], Viola97 [98]	Baker04 [6], Irani98 [37], Liu00 [50], Liu02 [51], Lucas81 [53], Sharma97 [89]
Focus	Baker04 [6], Bradley05 [11], Flusser99 [29], Lucas81 [53], Zitov99 [104]	Henn03 [34], Maes97 [54], Viola97 [98], Ward03 [100]	
Sensor	Henn03 [34], Irani98 [37], Liu00 [50], Liu02 [51], Maes97 [54], Sharma97 [89], Viola97[98]	Peng05 [78], Torkkola03 [95]	
Structure	Bookstein89 [9], Nielsen96 [13], Christensen96 [19], Crum04 [21], Agostino02 [22], McInerney96 [62], Moshfeghi91 [66], Rohde03 [83], Rueckert99 [84]		Brown03 [61], Gan04 [31], Henn03 [34], Lowe04 [52], Maes97 [54], Viola97 [98]

Table 3.1: A summary of the mapping of image registration algorithms (by reference) according to the forms of variation that they claim to support. Algorithms are placed within the five major dimensions of our taxonomy into three broad categories: algorithms invariant to that form of variation, algorithms that are capable at most problems with that form of variation, and finally algorithms that are barely capable of problems with that form of variation. Capable and barely capable mappings are made on the basis of results mentioned either within the corresponding paper or an equivalent paper with similar properties. This initial mapping provides insight into the capability of each image registration algorithms within the registration problem space.

Chapter 4

Modeling Image Registration

"There's no sense in being precise when you don't even know what you're talking about."

– John von Neumann

Image registration is the process of calculating spatial transforms which align a set of images to a common observational frame of reference, usually one of the images in the set. While some approaches to the problem are able to find a solution under a wide range of conditions, no single "best" image registration algorithm exists which solves all types of registration problems. There are many different algorithms that solve the problem in a variety of ways and perform differently under a variety of conditions. In Chapter 3 we introduced a taxonomy of image registration which mapped image registration algorithms according to the major dimensions of these variations. In this chapter we attempt to extend this mapping by providing a model of image registration which allows for the well defined specification of the conditions surrounding the problem and of the range of the desired solution, abstracting away from the functional specification of *how* to solve for a registration transform.

The goal when abstracting the image registration problem is not to provide a "black box" solution, but rather to provide a mechanism by which a description of the problem leads to the solution, with better descriptions leading to better solutions. Existing registration techniques and taxonomies have been examined in depth in order to to encapsulate the variations that allow the important aspects of the registration problem to be described. In creating a model which is based around these variations we have had to balance the flexibility of the model with ease of use and understandability.

A key aspect of our abstraction is that it be interpretable. In order to facilitate a clear interpretation of the model its components must be well

defined. In Appendix A we provide a formal *definition* of image registration and extend this definition into the applied domain. This formal definition uses set theory notation to specify the image registration problem. In Section 4.1 we explore the *representation* of the inputs and outputs used in the problem of image registration. Section 4.2 explores the different *conditions* of the problem of image representation, presented as forms of variation. Section 4.3 introduces the necessary concepts and types used in our model, and provides a mapping of the representations and conditions of registration into a formal model through which image registration can be expressed. Finally Section 4.4 expresses several common image registration problems under our model. These layers: Problem Definition, Problem Representation, Problem Conditions, and Problem Expression, represent our framework of accessible computer vision, and are presented below and explored in more detail in Chapter 8.

This model is a fundamental contribution of the thesis, providing a framework from which image registration problems can be described, and against which image registration algorithms may be measured. The n-dimensional problem space mapped by the model allows for the description of a problem, either as a point or as a volume, while the coverage and performance of a particular algorithm can similarly be modeled as a volume within this space.

A Registration Example: Panorama Stitching

As an introduction to the model we present an example panorama stitching registration problem in order to provide a context for the concepts explained below. The problem has been expressed relatively as the relationship between two images. Table 4.1 introduces a set of expectations and requirements for this problem: images are similar in exposure, aperture, etc, however they vary spatially. Exposure, aperture, and other properties are not specified below; properties which are not expressed are assumed to be the same under our model.

In our example we have specified that we expect that the images are overlapping by between 5% and 60%. We have also limited the solution space to transforms which result in an overlap between 5% and 50% based on the assumption that our dataset will be made up of images which vary by at least half an image.

Each of the concepts presented in the model are explained in detail below.

Image 1-2 Relative Expectations	Range / Value	Dist.
Overlap	[0.05,0.60]	+Quad
Image 1-2 Relative Requirements	Range	Dist.
Overlap	[0.05,0.50]	+Quad

Table 4.1: Example panorama stitching registration problem expressed as the relative relationship between a pair of images. +Quad = Quadratic Distribution

4.1 Representation of Image Registration Solutions

Every image registration problem is attempting to discover the spatial variation between two images. The two most common forms of spatial variation found in computational photography are stitching, where significant spatial variation occurs and images are being combined to expand the field of view, and stacking, where images are overlapping and are being combined to increase some other form of information such as dynamic range or focus. In addition, the extrinsic camera model affects the spatial alignment of images, and is listed below.

In order to account for spatial variation, a transform which aligns the images must be calculated. This transform provides a mapping from points in one image's coordinate system into the other's. The transform $\mathcal{A}$ from our definition of image registration represents this mapping and is a theoretical and exact mapping from one image space into the other.

4.1.1 Applied Representations

Applied transforms are most often represented using a 3×3 matrix, and a number of different transform types are used to solve for different problem conditions. Affine transforms are among the most common used in computational photography and medical imaging, and allow the rectangular image plane to be transformed into any parallelogram. The more flexible projective transform based representation is also used which allows the image plane to be transformed into any trapezoidal shape. These two transform types are outlined below.

Affine Transform

One of the most common representations of the solution space of image registration is the affine transform. Affine transforms approximate a planar mapping between two images through the relative translation, rotation, scale, and a skew which is used to approximate rotation in x and y. Translation refers to a horizontal (x) or vertical (y) shift between the two images within the representative image planes. Rotation refers to a rotation (in z) about the centre point or change in orientation between the two images within the representative image planes. Scale refers to a change in size (in both x and y) between the images. Finally skew refers to a difference in shear between the two image planes. Each of these forms of spatial variation is demonstrated individually below in Figure 4.1. Equation (4.1) demonstrates how an images' coordinates can be calculated using the matrix.

$$\begin{aligned} x' &= x(Scale\cos\theta + y(\sin\theta + X_{Skew}) + X_{Translation} \\ y' &= x(-\sin\theta + Y_{Skew}) + yScale\cos\theta + Y_{Translation} \end{aligned} \tag{4.1}$$

Perspective Transform

The perspective projection maps points in the three dimensional physical world onto an image plane using a set of projection lines which all meet at a single point referred to as the centre of projection. Perspective transformations define the relationship between two different projective planes, providing a mapping from one image plane into the other. In order to calculate the projection of one plane onto another a third parameter 'w', representing the distance of the point from the centre of projection must be solved for in addition to x and y. Each of the parameters specified for affine transformations can be expressed similarly in a perspective transform. Additionally the image can be rotated in the x and y dimensions, however the combination of these with the other operations into a single matrix is not easily expressible in an intuitive form. Figure 4.2 demonstrates a pair of images which differ by a perspective transform, and their corresponding alignment.

4.1.2 Extrinsic Camera Parameters

Changes of the position or orientation of the camera with respect to the scene also cause spatial variation, however they violate the assumption that the images correspond to a planar mapping of the same scene. If only

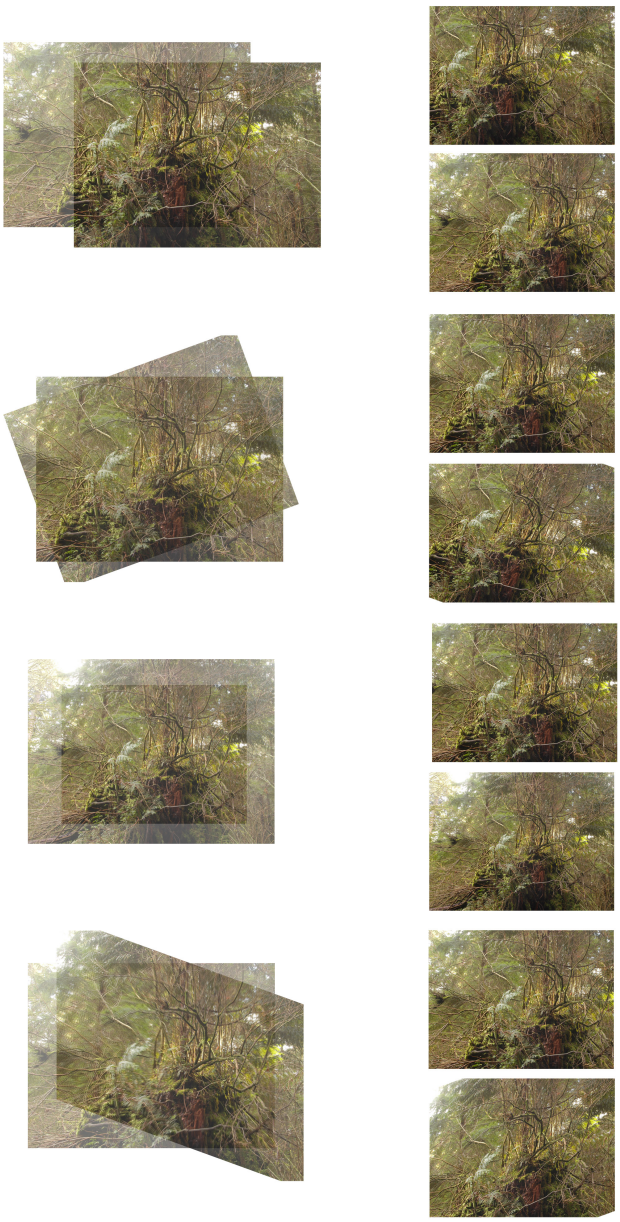

Figure 4.1: Individual elements that make up the affine transform. From top to bottom: Translation (both X & Y), Rotation, Scale, and Skew (both X & Y).

Figure 4.2: Demonstration of the perspective transformation

orientation changes then an affine or perspective transform can be used to solve for the matrix which aligns the image frames. As demonstrated by Figure 4.3, in the case of changes in camera position no transform can be found which will align the two images, although special cases such as perpendicular camera movement do exist. These images can however be aligned to a sparse (non-planar) 3d model of the scene using structure from motion techniques to create 3d photo collections[91]. Rough alignments are also possible for small movements of the camera with respect to its subject. The extrinsic camera parameters are those that define the position of the camera with respect to a particular frame of reference in the real world. This frame of reference is often taken from one of the images themselves when no known world frame is available. Extrinsic parameters define the camera's position in x, y, and z, as well as the orientation of the camera in the x, y, and z plane.

4.1.3 Overlap of Images

In many image registration problems the range of transforms which represent a possible solution space is unknown or is difficult to express as a range of possible transform parameters. In those situations however it is often still possible to estimate the amount of overlap between the two images. Particularly within the field of computational photography the overlap of image registration problems can be well estimated. Focus stacks, HDR image set, super-resolution image sets, and medical images should contain a high degree of overlap, while panoramas are likely to contain overlaps as high as 50% and as low as 5%. Figure 4.4 below highlights several image pairs which contain similar levels of overlap. Overlap can be calculated as

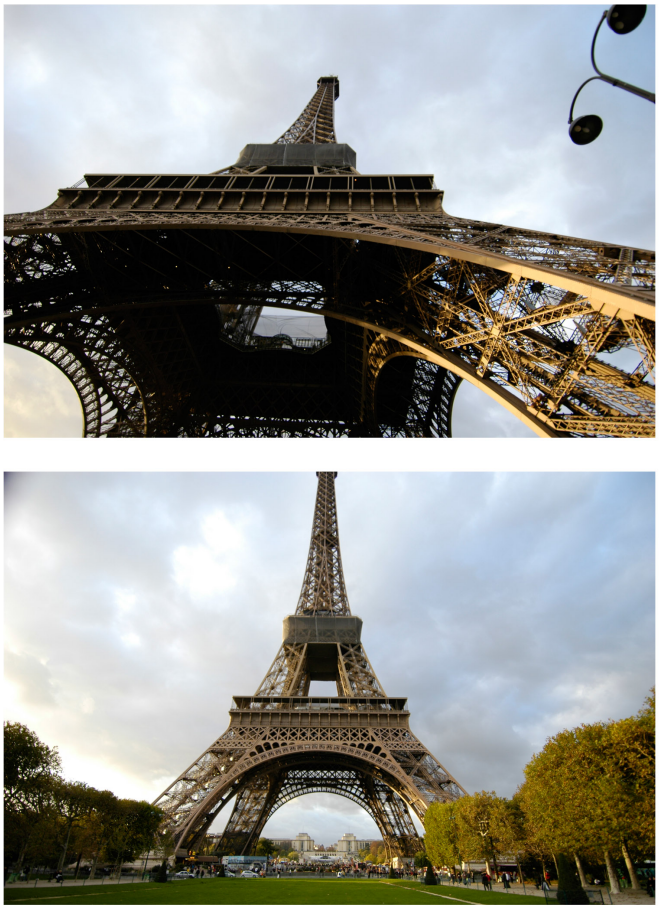

Figure 4.3: Eiffel Tower photographed from two camera positions. No affine or perspective transform can be found which will align these images because the positional extrinsic parameters of the camera are different for each image, violating the planar assumption of both the affine and perspective solution spaces.

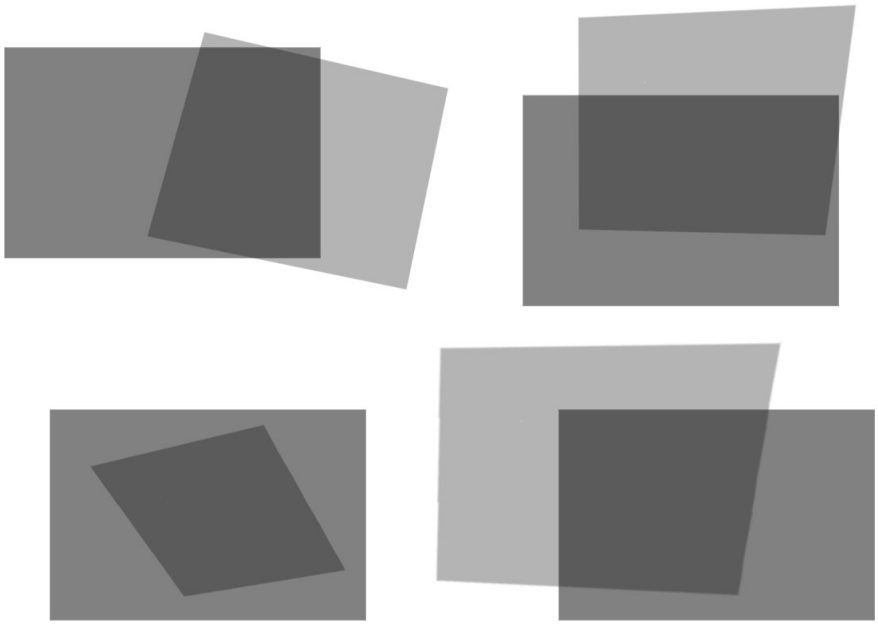

Figure 4.4: Four examples image pairs with similar levels of overlap. Overlap is calculated as the relationship between the number of overlapping pixels and the number of overlapping and non-overlapping pixels.

the relationship between the number of overlapping pixels and the number of overlapping and non-overlapping pixels.

In order to calculate overlap we calculate the area of the shape created by the intersection of the reference and synthetic images, dividing by the area of the reference to normalize. The vertices of intersection are calculated using the method developed by Toussaint [96] which finds the convex hull of intersection between the two images by determining whether a line from the point on one hull to the centre of the other hull intersects any lines. The area is calculated using Equation (4.2) which calculates the area of a polygon from known vertices; in our case between four and eight depending on the transform. x_n refers to the x value of the nth vertex, while y_n refers to its y value.

$$\frac{(x_1y_2 - y_1x_2) + (x_2y_3 - y_2x_3)... + (x_ny_1 - y_nx_1)}{2} \tag{4.2}$$

As we will see in Chapter 6, when we explore the impact that overlap has on the alignment of images, this concept is provides another useful abstraction of the image registration solution space. By evaluating the performance of image registration algorithms across a range of overlap we are able to better understand the limitations of particular algorithms and select appropriately when those conditions arise.

4.1.4 Algorithm runtimes

Another aspect of image registration problems which is sometimes important in the selection of algorithm is the length of time the algorithm takes to find a valid solution. For most algorithms a tradeoff can be made between performance and accuracy by varying parameter settings. Determining how long a given algorithm will take to come to a solution under a given set of parameters and images is a very difficult. Estimates based purely on algorithm are unreliable because image content can drastically affect runtime.

More accurate methods of estimation are necessary in order to begin to include this concept directly as a part of our model, however big O notation estimations of performance do give some indication of runtime, often highlighting parameters which may be limited or decreased in order to decrease it. The computation cost of one iteration of the Lucas-Kanade algorithm is $O(n^2N + n^3)$, where n is the number of warp parameters (6 for affine or 8 for projective), and N is the number of pixels in the template image T which is being aligned. The most expensive step by far is the computation of the Hessian, which alone takes time $O(n^2N)$. In the case of megapixel sized image alignment the image size becomes a dominant factor in the process, and subsampling the image, or using image pyramids can greatly reduce computation time. Rearranging the problem into an inverse compositional arrangement can also reduce the computation cost per iteration to $O(nN + n^3)$. To further limit the computation cost we can place an upper bound on the number of steps taken by the solver.

Feature based methods vary in computational cost, depending upon their feature calculation cost, the time taken to match features, and the time taken to solve for a universal solution which provides the best alignment. Reducing the number of features calculated is one method of drastically reducing the

computational cost, because the matching and solving stage are a function of this value. For example matching can be done using a k-d tree, which is $O(n \log n)$ per iteration, where n is the number of features. It is also possible to limit the maximum number of nearest neighbor checks to further reduce cost. Solving is frequently done using RANSAC [28], which is of order $O(nf)$ per iteration, where n is again the number of features, and f is the dimensionality of the feature. In order to further improve the computational cost the number of iterations of the algorithm can be reduced.

4.2 Conditions of Image Registration

The conditions under which image registrations occur have been explored in Chapter 3 through our novel taxonomy of image registration. Brown's concept of *variation* was extended, using it as suggested: as a basis for dividing the algorithms in the field. The variation, or differences, between two images form a basis for selecting an image registration algorithm. Table 4.2 revisits Brown's categorization of variations between images to be registered, and introduces these as specific conditions under which image registration commonly takes place. Also included in this table is the spatial variation referred to in our *representation* of image registration. Note that none of the variations listed in the table are tied to a particular algorithm or implementation.

Variation	Description	Representation
Corrected Distortions		
Translation	Misalignment due to translation	Proportional to the scale of geometry
Rotation	Misalignment due to rotation	Proportional to the scale of geometry
Scale	Misalignment due to scale	Proportional to the scale of geometry
Skew	Misalignment due to skew	Proportional to the scale of geometry
Camera Extrinsics	Misalignment due to changes in camera position	Position/Orientation of the Camera
Overlap	Overlap between images	Percentage of the reference image
Uncorrected Distortions		
Relative luminance	Luminance of the scene, relative to reference image	Proportional to the reference image
Focal Depth	Range over which images are sufficiently in focus	Distance (Hyperbolic Scale)
Scene Lighting	Changes in lighting between images	Descriptive Lighting Model
Camera Intrinsics	Model of the camera's intrinsic properties	Focal Length, Shutter Speed, Aperture, Lens Distortion, Sensor Model
Variations of Interest		
Object Models	Objects within the scene	Shape, Appearance
Object Motion	Amount of movement of objects between images	Local Motion Model

Table 4.2: Categorization of Variations between Images to be Registered. "Corrected distortions", "uncorrected distortions", and "variations of interest" are categories of variation specified in Brown [14].

These variations must be able to be specified by a programmer in order to describe the conditions of their registration problem, or by a researcher who is expressing the bounds of their algorithms' capabilities within the problem space. Following Brown's framework "corrected distortions" can be used to determine the type of transformation required, while "uncorrected distortions" and "variations of interest" can be used as the basis for selecting between registration methods.

Each of the dimensions of variation proposed in our taxonomy of registration are presented below. In Section 4.3 the individual properties that these forms of variation consist of are outlined in detail and their values, scale, and ranges are specified.

4.2.1 Intensity Variation

Intensity can vary between images for a number of reasons. Image pairs taken for high dynamic range (HDR) imaging have been deliberated manipulated to vary in *exposure* through changes in the intrinsic camera parameters, most often the shutter speed of the camera. This difference in exposure due to the change in intrinsic camera properties is the intensity variation we are attempting to provide a model for. Intrinsic properties which influence the exposure of the scene include aperture, shutter speed, and 'film' speed.

Other forms of variation often exhibit an effect on images which can cause changes in intensity within a pair of images. Pairs with varying illumination often also vary in intensity. A complex lighting model specifying the conditions of the lighting is possible through computer graphics representations, however we have yet to integrate this level of illumination specification into our model because of the complexity and the limitations of its usefulness in registration. Models for future computer vision problems, such as those mentioned in Chapter 8, may require such a model more directly. We provide a basic outline, using camera white balance settings as our basis, however its full specification and implementation is left as future work. Panorama sequences where light sources shine directly into the lens also represent another case where the intensity can vary significantly between a pair of images. Modeling this effect similarly requires a lighting model and is left as future work. Variations in sensor can also cause changes in both the intensity value of images as well as the gradient of the images. Our sensor model is outlined below in Section 4.2.3.

These other forms of variation should be specified directly when they are the cause of the changes in intensity in the image pairs. Specification of these variations as a variation in intensity can lead to incorrect classification

of the type of image registration problem, although as we will see in Chapter 7 these types of variation are often distinguishable from intensity variations directly by examining the images themselves.

Although variations in shutter speed are the most common source of intensity variations, many other intrinsic camera parameters can cause changes in variation. Below we outline the absolute properties of aperture, shutter speed, and film speed, which are associated with the luminance of an image. These values combine to form the overall exposure of an image. Assuming that the images are from the same scene as defined in Appendix A the calculation of the relative exposure of an image pair is straightforward. If an image pair is from a different scene then registration of that pair is an impossible task because an alignment between the two images does not exist.

Aperture

The aperture of a camera, or more accurately of a lens, is the opening through which light is exposed onto the film or sensor. Apertures are measured in f/stops, a fractional ratio of the diameter of the opening in comparison to the focal length of the lens. On a 50mm lens, an aperture of f/2 means that the diameter of the aperture is 25mm. The ratio is: $50/25 = 2$. The area of aperture can be calculated as:

$$area = \pi * r^2, \tag{4.3}$$

or in the case of our example, $\pi * 12.5^2 = 490.9\ mm^2$. The amount of light exposed onto the film or sensor is proportional to this area. In order to half the amount of light that the aperture allows, this area must be decreased by half. An aperture of 2.8 will have a diameter of 17.9mm, which when plugged into Equation (4.3) reveals an area of 250 mm^2. Table 4.3 outlines the relationship between traditional f/stop values and the corresponding area for a 50mm lens. As aperture values increase the size of the area and hence the amount of light let onto the sensor decreases. Doubling the amount of light is known as stepping up an f/stop or increasing the f/stop by a 'stop,' while halving the light is referred to as stepping down the aperture by a 'stop.' Since aperture values are expressed in proportion to the focal length of the lens the same amount of light reaches the sensor or film of a camera for a given f/stop, regardless of the lens. Figure 4.5 approximates two aperture settings, f/2.8 and f/22 and demonstrates how each lets light onto the film/sensor plane.

As we will see below in Section 4.2.2 changes in aperture also affect the depth of field of an image.

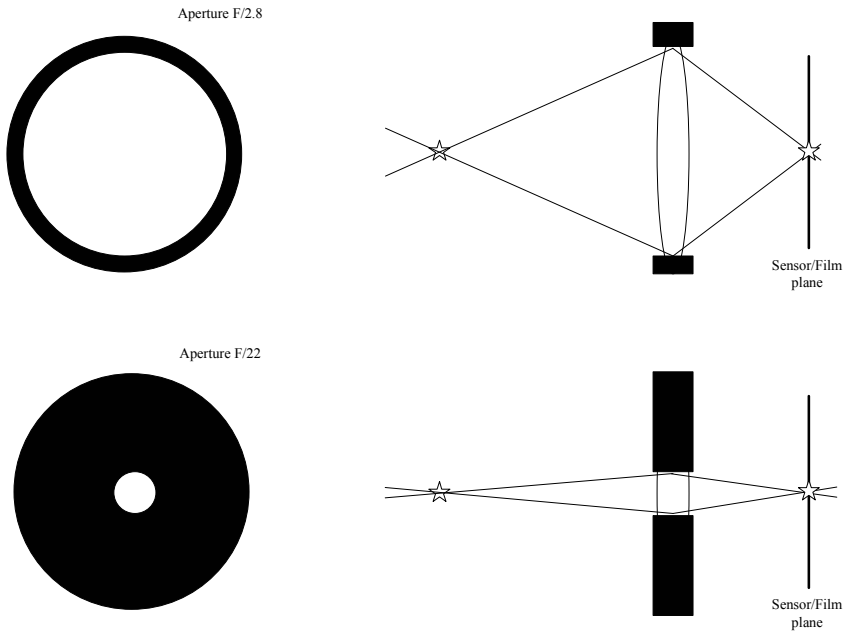

Figure 4.5: Aperture controls the amount of light that reaches the sensor. Smaller aperture values, measured in f-stops allow in more light.

Shutter Speed

Shutter speed is the amount of time that the sensor or film of the camera is exposed to light. For hand held photography the minimum desired shutter speed is usually the inverse of the focal length of the lens, thus a 300mm telephoto lens requires a minimum shutter speed of 1/300 of a second, while a 50mm lens requires only 1/50th of a second for the picture to contain no motion blur due to movement of the camera. The relationship between the shutter speed and the amount of light which reaches the film or sensor is direct: doubling the shutter speed allows twice as much light to reach the film/sensor plane. Table 4.4 demonstrates a sequence of shutter speed / f/stop combinations which allow the same amount of light to reach the sensor or film of a camera.

f/stop	Diameter (mm)	Radius (mm)	Area (mm^2)
f/1.0	50.0	25.0	1,963
f/1.4	35.7	17.9	1,002
f/2.0	25.0	12.5	491
f/2.8	17.9	8.9	250
f/4	12.5	6.3	123
f/5.6	8.9	4.5	63
f/8	6.3	3.1	31
f/11	4.5	2.3	16
f/16	3.1	1.6	8
f/22	2.3	1.1	4

Table 4.3: Aperture's F/stop in relation to area for a 50mm lens

Shutter Speed (s)	f/stop
1/4	f/45
1/8	f/32
1/15	f/22
1/30	f/16
1/60	f/11
1/125	f/8
1/250	f/5.6
1/500	f/4
1/1000	f/2.8
1/2000	f/2
1/4000	f/1.4

Table 4.4: Shutter Speed and Aperture combinations which allow the same amount of light to reach the sensor or film of the camera. When the shutter speed is halved the aperture must be stepped up a stop, or changed to a lower value, while a doubling of shutter speed requires the aperture be stepped down a stop, setting it to the next highest stop.

Film Speed

Film speed is the measure of a film's sensitivity to light. These values are determined by sensitometry and measured on various numerical scales, the most common being the *ISO* system. Insensitive film, with a low speed index requires more exposure to light to produce the same image as a more sensitive film, and is thus commonly termed a slow film. Highly sensitive films (those with a high ISO) are likewise referred to as fast films. A closely related ISO system is used to measure the sensitivity of digital imaging systems.

The relationship between film speed and the amount of light of a particular exposure is similarly direct, that is a doubling of the film or sensor's ISO will *effectively* allow twice as much light to reach the sensor since the sensor will be twice as sensitive to light.

Exposure Value

For the purposes of our model of image registration exposure values represent the amount of exposure that the sensor or film will receive when the shutter is opened. For a given scene under the same lighting conditions a change in these values will result in an illumination variation. Traditionally exposure values can be calculated as a combination of aperture, and shutter speed, however film speed affects the sensitivity of the sensor and has been included in our calculation of exposure values. Table 4.5 outlines the exposure values for several common lighting situations. Exposure value can be calculated using equation (4.4) [39], where N is the aperture value relative to f/1.0 and t is the exposure time in seconds. In equation (4.5) we factor in the addition of film or sensor sensitivity S, providing a relative value of the expected exposure that is independent of scene.

$$EV = \log_2 \left(\frac{N^2}{t} \right) \tag{4.4}$$

$$EV = \log_2 \left(\frac{N^2}{t} \right) + \log_2 \left(\frac{S}{100} \right) \tag{4.5}$$

A similar but more complex method of calculating exposure values which includes representations of the scene luminance and incident light luminance has also been examined [44]. The Additive System of Photographic Exposure combines: exposure time (shutter speed), aperture, exposure , ISO speed (film or digital imager sensitivity), Metered scene luminance (brightness), and Metered incident light illuminance (illumination).

For the problem of image registration the brightness Bv (measured in candelas/m^2) and incident illumination Iv (measured in lux) of the scene should remain constant, barring changes to the scene's lighting. As mentioned above we have chosen to include these types of variation within a scene lighting model from which these values can be derived. Equation (4.6) outlines how these values are incorporated into equation (4.5) to give a complete model of exposure. If no lighting model is provided then the scene is assumed to contain constant lighting and equation (4.5) can be used to calculate the exposure of the images as terms 3 & 4 of equation (4.6) will be the same for all images of that scene.

$$EV = \log_2\left(\frac{N^2}{t}\right) + \log_2\left(\frac{S}{100}\right) + \log_2\left(\frac{Bv}{3.4}\right) + \log_2\left(\frac{Iv}{67}\right) \tag{4.6}$$

4.2.2 Focus Variation

Focus in an image can vary for a variety of reasons. Many aspects of the camera's intrinsic properties affect the regions within the image which are considered to be in focus. When the aperture, or hole through which light is let into the sensor, changes the focal depth and focus of the image also changes. The focal length of the lens also affects this measure of focus. The distance of the subject from the lens plays a key role in focus as well, changing exponentially as the subject gets very close to the lens as is often the case in macro photography.

In photography the range of an image that is acceptably in focus is referred to as its depth of field [39]. Typically, photographers vary depth of field within an image by changing a camera's aperture, the size of the hole that lets light reach the sensor. Smaller holes let in less light, however the rays passing through the hole are less divergent, resulting in a greater depth of field. Figure 4.6 illustrates this principle.

The sharpness of an image is also potentially affected by both lens abberations and the diffraction limits of the sensor. Lens abberations occur at low f-stop values when light from a single point on the subject takes multiple paths through the optical chain, some of which do not converge to a single point on the sensor. Most single lens reflex (slr) camera lenses contain five or more optical elements and limit the amount of abberation in the optical pathway, however abberation in point and shoot cameras is common. We have chosen not to include abberation in our model of focus.

EV	TYPE OF LIGHTING SITUATION
-6	Night, away from city lights, subject under starlight only.
-5	Night, away from city lights, subject under crescent moon.
-4	Night, away from city lights, subject under half moon. Meteors (during showers, with time exposure).
-3	Night, away from city lights, subject under full moon.
-2	Night, away from city lights, snowscape under full moon.
-1	Subjects lit by dim ambient artificial light.
0	Subjects lit by dim ambient artificial light.
1	Distant view of lighted skyline.
2	Lightning (with time exposure). Total eclipse of moon.
3	Fireworks (with time exposure).
4	Candle lit close-ups. Christmas lights, floodlit buildings, fountains, and monuments. Subjects under bright street lamps.
5	Night home interiors, average light. School or church auditoriums. Subjects lit by campfires or bonfires.
6	Brightly lit home interiors at night. Fairs, amusement parks.
7	Bottom of rainforest canopy. Brightly lighted nighttime streets. Indoor sports. Stage shows, circuses.
8	Las Vegas or Times Square at night. Store windows. Campfires, bonfires, burning buildings. Ice shows, football, baseball etc. at night. Interiors with bright florescent lights.
9	Landscapes, city skylines 10 minutes after sunset. Neon lights, spotlighted subjects.
10	Landscapes and skylines immediately after sunset. Crescent moon (long lens).
11	Sunsets. Subjects in open shade.
12	Half moon (long lens). Subject in heavy overcast.
13	Gibbous moon (long lens). Subjects in cloudy-bright light (no shadows).
14	Full moon (long lens). Subjects in weak, hazy sun.
15	Subjects in bright or hazy sun (Sunny f/16 rule).
16	Subjects in bright daylight on sand or snow.
17	Rarely encountered in nature. Some man made lighting.
18	Rarely encountered in nature. Some man made lighting.
19	Rarely encountered in nature. Some man made lighting.
20	Rarely encountered in nature. Some man made lighting.
21	Rarely encountered in nature. Some man made lighting.
22	Extremely bright. Rarely encountered in nature.
23	Extremely bright. Rarely encountered in nature.

Table 4.5: Exposure Values for various lighting conditions at ISO 100 [76].

Conversely at high f-stop values, light passing through smaller holes is normally considered to be less divergent, however diffraction begins to become an issue as the circumference of the small hole admitting light causes a greater number of light rays to interfere with each other [39]. If this diffraction pattern spreads beyond the width of a pixel in the sensor, the resulting image will be less in focus. When values for colour are calculated using the bayer pattern [39] the diffraction limit is usually considered to be the f-stop that causes an interference pattern with a width of two pixels. This limit typically falls around f/22 for digital single lens reflex (DSLR) cameras, but can be as low as f/5.6 for smaller point and shoot cameras because of their small pixel size. Photographing at f-stop values higher than the diffraction limit results in an image that is less focused over the entire image, however it does not affect the image's depth of field.

Circle of Confusion

When a pair of images shares no overlapping regions which are 'in focus' then they are difficult to align by feature based methods, and other methods which are focus invariant must be selected in order to align the pair. This common notion of a part of the image being 'in focus' or 'out of focus' is a significant simplification of the concept of focus. In fact *how* 'in focus' a part of the image is can be calculated. If a part of the image is considered to be 'in focus' it means that part is sufficiently in focus: the focus value is greater than some specific value. This threshold value depends on the size and shape of pixels on the camera's image sensor, the resolution of the printer, as well as the distance from which an image is meant to be viewed.

Measures of focus center around the concept of a circle of confusion, which represents the smallest circle that a person can distinguish from a specific distance. Focus within an image is a continuum, and the circle of confusion measure acts as a threshold; details in the image that meet the criteria are considered to be "in focus," while those that don't are labeled "out of focus." In digital photography this value is set to the size of a pixel on the camera's sensor, as that is the smallest element which will be resolved in the digital image.

Aperture

The size of the aperture, or hole, through which light is let into the sensor plays a key role in how much of the image will be in focus. Figure 4.6 demonstrates how smaller aperture values have a narrower depth of field,

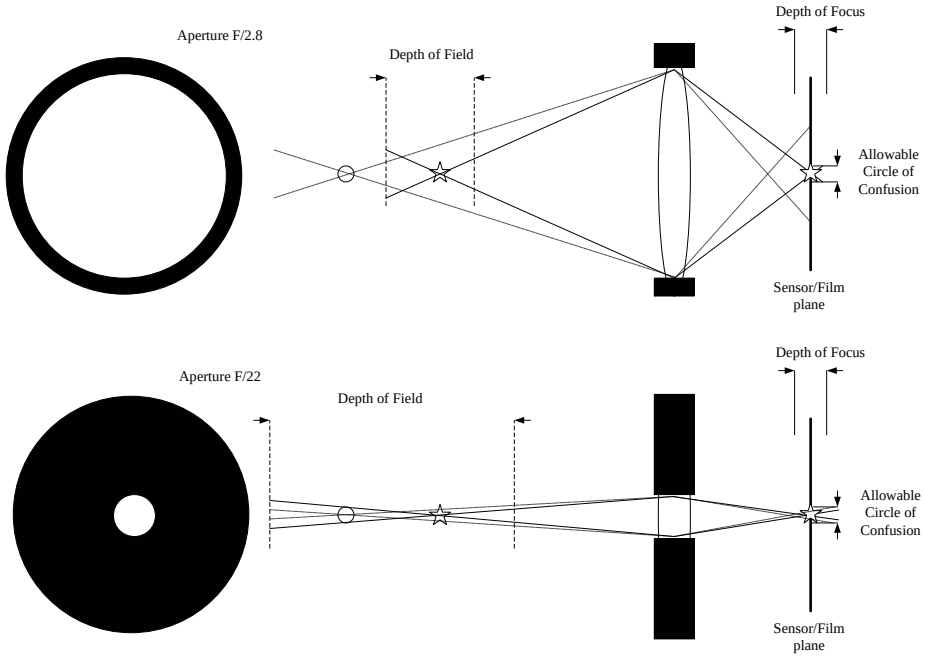

Figure 4.6: In addition to controlling the amount of light which reaches the sensor, aperture also controls the depth of field of the image. Smaller aperture values have a narrower depth of field, while larger values have a deeper depth of field.

while larger values have a deeper depth of field. Figure 4.7 demonstrates this principle photographically.

Subject Distance

The depth of field of the image, as well as the region of the image which is sufficiently in focus is dependent upon the distance of the subject being focused from the camera. Focus on closer subjects result in a narrower depth of field, while subjects at a greater distance have a greater depth of field.

Hyperfocal Distance

The hyperfocal distance is the nearest focus distance at which the depth of field of the image extends to infinity; the sensor or film is no longer

Figure 4.7: In addition to controlling the amount of light which reaches the sensor, aperture also controls the depth of field of the image. Smaller aperture values have a narrower depth of field, while larger values have a deeper depth of field. The image on the left has an aperture of f/2 while the one on the right has an aperture of f/8.

capable of rendering detail beyond this distance. Focusing the camera at the hyperfocal distance results in the largest possible depth of field for a given f-stop . Focusing beyond the hyperfocal distance does not increase the far depth of field but rather decreases the depth of field in front of the subject, effectively decreasing the total depth of field. The hyperfocal distance can be approximated as:

$$H = \frac{F^2}{F\#C} \tag{4.7}$$

where F is the focal length of the lens (in mm), F# is the aperture value of the camera (in F-stops), and C is the size of the circle of confusion (in mm).

Depth of Field

The majority of professional photographers shoot in a mode known as aperture priority, which fixes aperture at a specific value and dynamically adjusts the exposure time (shutter speed) to allow in the amount of light required by the sensor to properly expose the image. By fixing the aperture, the depth of field of the images is also relatively fixed, affording more control over the image.

The approximate equation governing depth of field and exposure is given below:

$$DOF = \frac{2SD^2F\#C}{F^2} \tag{4.8}$$

where SD is the subject distance (in mm) F# is the aperture value of the camera (in F-stops), C is the size of the circle of confusion (in mm) and F is the focal length of the lens (in mm).

For a subject which is 2500mm from the camera, photographed with an aperture of F2.8 on a digital camera with a pixel size of 0.15 and a lens of focal length 50mm the depth of field is calculated as:

$$DOF = \frac{2 * 2500^2 * 2.8 * 0.15}{50^2} = 14m$$

A subject which is 50mm from the camera photographed with the same properties:

$$DOF = \frac{2 * 50^2 * 2.8 * 0.15}{50^2} = 0.84mm$$

has a much lower depth of field of 0.84mm.

In Focus Distances from an Image

The minimum distance from the camera which will be sufficiently in focus can be approximated by subtracting half of the depth of field from the subject distance. Similarly the maximum subject distance which will be sufficiently in focus can be approximated by adding this value. While this approximation works for lenses of 100mm focal length and above, the distribution of front and rear depth of field is not always symmetric. Table 4.6 outlines the distribution of depth of field for lenses of a variety of focal lengths.

Using this table as a basis, we are able to estimate with reasonable accuracy the minimum and maximum distance from the camera which will be in focus. This allows us to compare these values across different images, allowing our model to express focus variation.

Focus Value

In an attempt to make focus variations easier to specify we have developed an abstraction of focus known as a focus value which can be calculated from the in focus distances of two images. This value is a representation of the degree to which two images contain overlapping in focus regions, and is measured as the ratio of the sum of the depth of field (DOF) in each image

	Distribution of depth of field	
Focal Length (mm)	Rear	Front
10	70.2 %	29.8 %
20	60.1 %	39.9 %
50	54.0 %	46.0 %
100	52.0 %	48.0 %
200	51.0 %	49.0 %
400	50.5 %	49.5 %

Table 4.6: Approximate distributions of depth of field from rear to front for different lens focal lengths.

to twice the distance between the nearest and furthest in focus pixel from both images. Equation (4.9) demonstrates the method of calculating our focus value.

$$Focus = \frac{DOF_1 + DOF_2}{2(\max(far_1, far_2) - \min(near_1, near_2))} \tag{4.9}$$

Under this abstraction image pairs with entirely overlapping in focus distances have a focus value of 1.0, while pairs whose in focus distances connect but do not overlap have a focus value of 0.5.

4.2.3 Sensor Variation

Sensor variations occur in a variety of situations most often identified with particular research areas. In medical image processing images taken with different sensors are combined in order to provide a more detailed understanding of the patient or subject. The algorithm used in the alignment of these images is dependent upon the type of sensor data. When sensors vary the gradient direction, intensity representation, and range of possible values all change. In this section we provide an initial and very basic model of sensors in order to allow for the expression of problems which are sensor varying.

Remote sensing also commonly features images taken with different modalities, allowing for more sophisticated analysis of a scene or environment. As with medical imaging these multimodal images require image registration algorithms which are invariant to the change in sensor.

Finally, sensor variation can also occur when different cameras of the same sensor type, but with different sensor properties are used to capture views from the same scene. In this case the gradient and intensity repre-

Sensor Type
Film
CCD
CMOS
Infrared
Magnetic Resonance Imaging (MRI)
Electron Microscopy
Ultrasound
Projection Radiography
Fluoroscopy
Photoacoustic Imaging
Tomography
X-radiation (XRAY)
Electroencephalography (EEG)
Magnetoencephalography (MEG)
Electrocardiography (EKG)
Positron Emission Tomography (PET)
Interferometric Synthetic Aperture Radar

Table 4.7: Types of sensors available under our simplified model of sensor variation.

sentations are unlikely to change, however the sensor size, pixel size, colour model, etc. all play a significant role in the capture of an image, and knowing these differences a priori can be beneficial when mapping from one image space into another. In most cases this information can be derived directly from the intrinsic camera model if available.

Sensor Modality

Fundamental to the determination of whether two images are sensor varying is the type of sensor used. While future sensor model may provide a mechanism for converting between sensor types, we use a simple list of possible sensor types, requiring that these match between images in order for an image to not be sensor varying. This approach is crude, but allows for the expression of sensor variation in instances where this is critical to the selection of appropriate algorithms to align the images. Table 4.7 lists the types of sensors available in our model. A more sophisticated sensor model is left as future work for researchers who are focused on medical imaging or remote sensing.

4.2.4 Scene Variation

Scene variations occur between pairs or sets of images for a number of reasons. Sets of images from a scene are by necessity captured over time, allowing for changes in the scene between images. In photography this frequently occurs as people or objects in a scene move between images. In medical imaging these variations can be growths in tumors or changes in muscle mass. We have grouped these forms of variation into two distinct categories: Local motion of objects between capture, and deformation of objects within the scene.

In addition to changes in the scene *between* images, if exposure times are not fast enough to capture the scene both local (subject) and global (camera) motion are possible within an individual image. These latter two forms of scene variation are within individual images however, and are not included in our model. Development of a global motion model to compensate for camera blur is feasible, but not applicable to image registration. A local model of motion is much more complex and requires more thought regarding the representation of the subject and its range of possible motion more suited to object tracking.

Local Motion Between Capture

As mentioned above, a sophisticated local model of motion is much more complex and requires more thought regarding the representation of what is moving and its range of possible motion more suited to object tracking. However for the task of image registration, knowing that some objects in the scene are moving can potentially allow for more leniency in the global matching of segments.

Our model for local motion between capture focuses on this simpler description, allowing the expression of the percentage of the image which has changed between images, and the distance that objects have moved, expressed in relation to the scene.

Deformation

Deformation of an object within the scene is common within medical imaging. Again, if this is to be modeled with sophistication, a representation both of the object, and of the possible range of changes over time is necessary. The development of such a model is outside the scope of this thesis and is left as future work for researchers within the medical imaging community who have a deep understanding of the objects being imaged.

Our model for deformation is simpler, once again allowing the expression of the percentage of the image which has changed between images, and the amount that objects have changed, expressed in relation to the original size of the object. This provides a very crude basis for describing deformation that makes developers of image registration algorithms aware of these conditions between the images. If these deformations are the desired output of the system, then the representation of the transform must change to one that solves for local spatial alignments between pairs, either on a per pixel basis as in optical flow, or using a grid of affine or perspective transforms.

4.3 The Expression of Registration

In Appendix A and Section 4.2 we have defined the problem of image registration. In order to create an interpretable model of image registration we must provide a means of specifying both the desired representation of the problem, as well as the conditions under which the registration is occurring.

In this section we introduce our model of image registration and reexamine our example of the expression of a panorama stitching problem. To create a model for image registration a number of basic concepts must first be defined. When combined with our definition of representation and conditions, these concepts allow for the creation of an expressive model of image registration. Primarily these concepts deal with the specification of value or difference of the various parameters that make up the image registration problem space.

4.3.1 Expectations and Requirements

In order to allow a greater degree of flexibility in the description of vision problems we allow application developers to specify the conditions of the problem and limitations of the solution both in terms of their *expectations* and their *requirements*. Expectations are conditions or limitations that are believed likely (are expected) to occur, while requirements are hard limits on either the input conditions or on the solution space. The differences in interpretation of expectations and requirements by our interpreter is detailed in Chapter 6.

Expectations

Expectations are suppositions that a developer has on the conditions of the problem or limitations of the solution space. These are not hard or limiting in any way, however, knowing these allows the interpreter to determine which algorithms are going to be appropriate to use when finding a solution. As an example, the expectation that the exposure has changed between two images can guide in the selection of exposure invariant algorithms. This process of describing what kind of information a developer is expecting is meant to mimic the expert knowledge that a vision researcher uses when selecting an algorithm.

Requirements

Requirements are limitations that the developer wishes to place on either the input conditions of the problem, or more likely on the solution space to the vision problem being solved. These limitations are considered 'hard' by the system and any detectable conditions or solutions that don't meet these specifications are considered to be invalid. By way of example a developer, knowing that they are doing a high-dynamic-range image stack can require that the resulting registration transform produced by OpenVL contain only very small changes in rotation or scale or an overlap of at least 90%. This requirement can be used to both guide in the selection of algorithms that use constraints based solvers, and in the selection of a solution in the instance where multiple solutions have been produced.

4.3.2 Properties

In order to describe both the expectations and the requirements of the problem space the developer must be able to express the conditions and the representation or solution of their problem. In most cases these conditions and representations can be expressed through *properties*; the aspects of an image that can be specified as a value or range of values. Each property has a specific unit of measure and scale which can be compared relatively across image pairs or sets.

Using the automatic classification techniques described in Chapter 7, EXIF information, Dicom metadata, or camera specific metadata formats it is also possible to automatically fill in many of the properties and models of a given image, particularly those related to camera intrinsics. The image registration model, however, has been designed to be flexible enough to deal with cases where this information is not available, allowing the application

developer to specify the conditions of their problem manually, and vision researchers to be able to specify the conditions under which their algorithms work in a well defined manner.

Properties are any aspects of an image, image pair, or image set that can be expressed using a numerical value. There are two main types of model properties: *absolute* and *relative*. Absolute properties are those which can be calculated or measured as a single numerical value or range of values. They represent a property of the image or scene. Working from the premise that it is possible that a developer will not know the exact value of given properties, we allow the specification of properties using distributions. These distributions express the range of allowable or suspected values for the property and can add a bias to the representation.

A Uniform distribution is the default, giving every value within the specified range an equal likelihood, but under our current framework we also support Exponential, +Quad & ∪-Quadratic, Gaussian and Inverse Gaussian distributions allowing for much more complex descriptions of the problem. This variety is designed to provide flexibility in design and development of vision application. In practice the specification of these distributions could be done through easy to use macros or developers who don't require this flexibility could rely on default values. Absolute properties can also be expressed directly as a histogram, allowing developers to describe or calculate a property with a biased range of expected values. Examples of absolute properties include intrinsic camera parameters such as shutter speed or ISO, Luminance, and the individual parameters of the transform which aligns the image pair. Table 4.8 outlines the parameters which must be specified for each type of distribution.

For image registration the variation between images defines the main dimensions of the problem space. *Absolute* properties specify the properties of a single image, such as the aperture or shutter speed of the camera, the luminance of the image, or the focus of the image. *Relative* properties specify relationships between an image pair, such as the translation, rotation, scale, or skew of an affine transformation, the relative luminance or the relative focus. Relative properties may either be differences or ratios depending upon the property.

As we saw above in Section 4.2 some of the properties in our model of image registration are calculated using equations which rely on other properties, summarizing multiple properties into a single range of values for ease of expression and understanding. It is much simpler, for example, to specify the exposure of an image, rather than specifying the shutter speed, aperture, and ISO, from which the exposure can be calculated. Similarly the

Distribution	Equation
Uniform	$f(x\|a,b) = 1$, for $x \in [a,b]$, 0 otherwise
Inverse Uniform	$f(x\|a,b) = 0$, for $x \in [a,b]$, 1 otherwise
-Quadratic	$f(x\|a,b,\alpha,\beta) = \alpha(x-\beta)^2$, for $x \in [a,b]$; $\alpha = \frac{12}{(b-a)^3}, \beta = \frac{b+a}{2}$
+Quadratic	$f(x\|a,b,\alpha,\beta) = \frac{(\beta-x)^2}{\alpha}$, for $x \in [a,b]$; $\alpha = \frac{12}{(b-a)^3}, \beta = \frac{b+a}{2}$
Exponential	$f(x\|\mu,\lambda) = \lambda e^{-\lambda x}$
Gaussian	$f(x\|\mu,\sigma) = \frac{1}{\mu}\phi(\frac{x-\mu}{\sigma})$
Inverse Gaussian	$f(x\|\mu,\sigma) = \frac{\mu}{\phi(\frac{x-\mu}{\sigma})}$

Table 4.8: Distribution types supported in the expression of properties.

overlap is easier to express than the combination of all possible transform ranges. In most cases the equations of our model are designed to take advantage of available metadata embedded within images, however they work equally well with properties which have been specified as distributions by the developer. If metadata is available for a property, then that property is specified as a uniform distribution with a range of 0. In the case where a derivable property has been specified, the specified value is used.

We also allow the creation of *sets* of images. In order for the interpreter to be able to properly select image pairs from the set and understand the relationship between those images, sets of images require that every pairwise relative property be known. For value properties this can take the form of a specification of properties relative to a single reference image, from which the unspecified properties can be calculated. Relative range properties require a complete mapping of all possible pairwise combinations. For convenience we provide functionality within our model to test the completeness of a set.

4.3.3 Absolute Properties

Absolute properties, both values and ranges, are represented by a distribution. The specifications of the aspects of particular distribution types is outlined above in Table 4.8. Because they are estimates, absolute properties

can also be represented by multiple distributions which combine with each other into a mixture of distributions. This added flexibility allows developers to express more complex representations of a given property. The specification of the absolute properties used in our model of image registration is discussed in Section 4.3.

We also define the allowable ranges of the properties of our image registration model. Application developers do not need to supply a complete model, only supplying those classifications and constraints that are important to their problem. When values are known they are specified using a Uniform distribution with a delta of zero, as is done in the case of values known or calculated using metadata.

4.3.4 Relative Properties

When combining two images into a pair their relative properties are either expressed as a distribution or calculated as the integral of the two absolute properties of the images. A relative difference property is calculated as the possible distribution of differences between the two absolute properties of the image pair. The relative relationship between properties may also be represented as a ratio. Relative ratio properties are calculated as the ratio of the distributions of the absolute property, as this intersection describes the places where the ranges of the images overlap. This relative ratio property is calculated using the reference (or first) image as the denominator.

In order to facilitate easier description relative properties can either either be set by combining distributions specified as absolute properties at the individual image level, or by specifying the relative property directly at the pair level. For sets of images every pairwise combination of relative properties must be known in order to properly interpret the set. Thus for a set of N images, N! relative properties must be either specified or calculated. When an image is added to a set which has been specified absolutely the relative properties can be calculated directly from these values. If the developer chooses to specify the relationship between the images relatively a more complete specification is required. For ratio properties the relative properties can all be specified in relation to a single *reference image*, from which the other unspecified properties can be calculated. In order to specify difference properties relatively we require all pairwise combinations of the images to be specified as the differences cannot be calculated relative to a single reference. For convenience we provide functionality within our model to test the completeness of a set of images.

4.3.5 Belief

In the case of expectation, a belief value can be used to weight the combination of distributions, allowing the developer to specify that a particular range is more or less likely. Weighted distributions are combined and then normalized so that the total likelihood sums to 1.

4.3.6 Categories

In a number of instances a more complex model for a particular condition or representation is possible, but not necessary for the problem of image registration. In those cases a simple category based model is used to provide a crude means of expression, allowing later researchers who are more invested in the model to create it.

Absolute categories are specified according to whichever category an image corresponds to. Relative categories are expressed as the categories of both of the images in the corresponding image pair, essentially creating categories for every categorical combination. This is done to allow an algorithm with the ability to convert or prepare the images for image registration to be selected by the interpreter, and to properly set up the problem.

4.3.7 A Model for Image Registration

From the conditions and representation specified above we have developed a model to express image registration problems. In the problem of image registration the variation between the properties of images to be aligned provides the context through which appropriate algorithms may be selected. As mentioned above the properties can be specified absolutely for individual images, or relatively for image pairs. For sets of images all pairwise combinations must be calculable either through a reference image or through specification of the complete set. Table 4.9 provides the model for the expression of the properties of a single image relevant to image registration. In Table 4.10 these properties are relative, compared between pairs of images, and a representation of the range of possible transforms between the pair can be expressed. These relative values can either be specified directly or derived from the absolute properties of each image. In the creation of a set of images all pairwise combinations of relative properties must specified or derived. In Section 4.4 each of these means of problem expression is explored using a variety of image registration problems.

The models introduced in Tables 4.9 and 4.10 can be specified as both expectations and as requirements, allowing the expression of both possible

Conditions	Units	Model Type
Exposure	EV	AP
Region of Focus	Range of Distance (in m)	AP
Scene Lighting		AM
Intrinsics.ShutterSpeed	Seconds	AP
Intrinsics.Aperture	FStops	AP
Intrinsics.FilmSpeed	ISO Units	AP
Intrinsics.FocalLength	Millimetres	AP
Intrinsics.DepthOfField	Range of Distance (in m)	AP
Intrinsics.SubjectDistance	Distance (in metres)	AP
Intrinsics.CircleOfConfusion	Diameter (in mm)	AP
Intrinsics.Sensor	Type	AC
Region of Focus	Range of Distance (in m)	RP
Extrinsics.Position.X	Distance (in m) from Scene Origin	RP
Extrinsics.Position.Y	Distance (in m) from Scene Origin	RP
Extrinsics.Position.Z	Distance (in m) from Scene Origin	RP

Table 4.9: Absolute Image Model for the expression of the conditions of a single image. No representation of transformation is possible with a single image. A = Absolute, R = Relative, P = Property, M = Model, C = Category

Representation	Units	Model Type
Translate.X	Image Widths	RDP
Translate.Y	Image Heights	RDP
Scale.X	Image Size	RTP
Scale.Y	Image Size	RTP
Rotation.Z	# of Clockwise Revolutions	RDP
Rotation.X	# of Clockwise Revolutions	RDP
Rotation.Y	# of Clockwise Revolutions	RDP
Skew	Image Widths	RDP
TransformDensity.X	Num per Image Width	AP
TransformDensity.Y	Num per Image Height	AP
Conditions	Units	Model Type
Δ Exposure	EV	RDP
Δ Region of Focus	Range of Distance (in m)	RDP
Δ Scene Lighting		RDM
Deformation	Percentage	RTP
Scene Movement	Percentage	RTP
Intrinsics.ShutterSpeed	Seconds / Seconds	RTP
Intrinsics.Aperture	FStops / FStops	RTP
Intrinsics.FilmSpeed	ISO Units / ISO Units	RTP
Intrinsics.FocalLength	Millimetres / Millimetres	RTP
Intrinsics.DepthOfField	Range of Distance (in m) -	RDP
Intrinsics.SubjectDistance	Distance (in m)	RTP
Intrinsics.CircleOfConfusion	Diameter (in mm)	RTP
Intrinsics.Sensor	Type	RC
Extrinsics.Position.X	Distance (in m)	RTP
Extrinsics.Position.Y	Distance (in m)	RTP
Extrinsics.Position.Z	Distance (in m)	RTP
Overlap	Percentage	RTP
Focus	Focus Value	RTP

Table 4.10: Relative Pairwise Model for the expression of the registration of a pair of images. The necessary representation and possible conditions surrounding registration of the two images can be specified or derived using this model. A = Absolute, R = Relative, D = Difference, T = raTio, P = Property, M = Model, C = Category

and necessary representations and conditions of an image registration problem. The model is extensible, but hopefully provides enough coverage in its initial form to allow for the mapping of all known forms of image registration problems. In Chapter 6 we will explore the interpretability of the model.

4.4 Example Image Registration Problems

In this section we explore the expression of common image registration problem classes, including: panorama stitching, focal stacking, high-dynamic-range imaging, multimodal imaging, and super-resolution imaging. Problems can be expressed either as a combination of individual images, as pairs, or as sets, and we examine each of these types within the examples below. Additionally both expectation and requirements can be specified, allowing a full range of expression of the image registration problem space. The interpretation of the expectations and requirements placed on the problem conditions and representation can be found below in Chapter 6.

In all of our examples, the geometry of images are defined using a scale of between [0.0,1.0] on each axis. Thus a translation of 0.5 between two images translates the second image so that its centre is at the edge of the first. Rotations, Scales, and Skews are similarly defined on a [0.0,1.0] scale: 1.0 represents a full clockwise rotation, while -0.25 represents a 90 degree counterclockwise rotation; 0.0 difference in scale representing an image of the same size, and -1.0 representing an image half as large (a difference in size of 1.0); and 0.25 representing an image skewed by a quarter of the image's width respectively.

There are many ways of determining the model values including problem analysis, visual analysis of image pairs, and use of metadata to calculate model values directly. We present examples of each below as we demonstrate our model.

4.4.1 Panorama Stitching

As an introduction to the model we present an example panorama stitching registration problem in Figure 4.8. The problem has been expressed relatively as the relationship between two images. Table 4.11 introduces a set of expectations and requirements for this problem: images are similar in exposure, aperture, etc, however they vary spatially. Exposure, aperture, and other properties are not specified below; properties which are not expressed are assumed to be the same under our model.

Figure 4.8: Input images which contain only spatial variation.

In our example, based on a visual inspection of the images, we have specified that we expect that the images are overlapping by between 35 & 5%. This broad range is meant to mimic the uncertainty of non-experts who are specifying problems using the model. We have also directly specified that there is no change in exposure and that the in focus regions of both images is the same.

Image 1-2 Relative Expectations	Range / Value
Overlap	[0.05,0.35]
Δ Exposure	0 EV
Focus Value	1.0

Table 4.11: Example panorama stitching registration problem expressed as the relative relationship between a pair of images. Distribution

The interpretation of this model and the selection of an appropriate algorithm with which to find a solution is discussed in detail in Chapter 6.

4.4.2 Focal Stacking

For our focal stacking example below in Table 4.12 we express the absolute properties of three images individually, allowing for the creation of a set of images where the relative properties are calculated rather than specified. Figure 4.9 demonstrates two of the images from the set. Each image is defined by its intrinsic properties of shutter speed, aperture, film speed, focal length, subject distance, and circle of confusion. In this example these values were derived or calculated directly from exif metadata that was included

with the images when the photograph was taken. From these values the absolute values of exposure, depth of field, and in focus can be calculated. The relative properties of each pairwise combination can be calculated from these absolute values. Table 4.13 demonstrates the relative properties calculated between each possible pair. These absolute and relative properties are expressed as expectations that we have regarding the conditions which surround the problem.

Figure 4.9: Input images which contain focal variation.

Notice how all of the properties in this example were of fixed value, assumed to be derived from exif information. In the next example we will examine how absolute properties expressed as distributions are integrated into their corresponding relative difference and ratio properties.

4.4.3 High-Dynamic-Range Imaging

For our high-dynamic-range we express a case where the system using our model must be flexible in order to deal with a variety of input conditions. Exact exposure values for the images are not known, requiring specification of the expected conditions using a range of possible values. The expectations are expressed as absolute properties in Table 4.14 and are used to calculate the relative expectations derived in Table 4.15. The range of possible transforms is also expressed as a requirement of this problem, limiting the solution space to that common to a stacking problem. The requirements on the representation are specified below in Table 4.16. Figure 4.10 shows images one and two from the example.

Image 1 Expectations	Value	Dist.
Intrinsics.ShutterSpeed	1/250 Second	U
Intrinsics.Aperture	F2.8	U
Intrinsics.FilmSpeed	100 ISO	U
Intrinsics.FocalLength	50 mm	U
Intrinsics.SubjectDistance	0.8	U
Intrinsics.CircleOfConfusion	0.15 mm	U
Calculated.Exposure	11 EV	U
Calculated.DepthOfField	215 mm	U
Calculated.InFocus	[692.48, 907.52] mm	U
Image 2 Expectations	Value	U
Region of Focus		U
Scene Lighting		U
Intrinsics.ShutterSpeed	1/250 Second	U
Intrinsics.Aperture	F2.8	U
Intrinsics.FilmSpeed	100 ISO	U
Intrinsics.FocalLength	50 mm	U
Intrinsics.SubjectDistance	1.0	U
Intrinsics.CircleOfConfusion	0.15 mm	U
Calculated.Exposure	11 EV	U
Calculated.DepthOfField	336 mm	U
Calculated.InFocus	[832, 1168] mm	U
Image 3 Expectations	Value	
Intrinsics.ShutterSpeed	1/250 Second	U
Intrinsics.Aperture	F2.8	U
Intrinsics.FilmSpeed	100 ISO	U
Intrinsics.FocalLength	50 mm	U
Intrinsics.SubjectDistance	1.60	U
Intrinsics.CircleOfConfusion	0.15 mm	U
Calculated.Exposure	11 EV	U
Calculated.DepthOfField	860 mm	U
Calculated.InFocus	[1169.92, 2030.08] mm	U

Table 4.12: Expression of three images that are part of a focal stack. Absolute properties are specified. U = Uniform Distribution

Image 1-2 Relative Expectations	Value	Dist.
Δ Exposure	0 EV	U
Range of Focus	[832,907] mm	U
Intrinsics.ShutterSpeed	1.0	U
Intrinsics.Aperture	1.0	U
Intrinsics.FilmSpeed	1.0	U
Intrinsics.FocalLength	1.0	U
Intrinsics.DepthOfField	121 mm	U
Intrinsics.SubjectDistance	0.2 m	U
Intrinsics.CircleOfConfusion	1.0	U

Image 2-3 Relative Expectations	Value	Dist.
Δ Exposure	0 EV	U
Range of Focus	0 mm	U
Intrinsics.ShutterSpeed	1.0	U
Intrinsics.Aperture	1.0	U
Intrinsics.FilmSpeed	1.0	U
Intrinsics.FocalLength	1.0	U
Intrinsics.DepthOfField	-524 mm	U
Intrinsics.SubjectDistance	0.6 m	U
Intrinsics.CircleOfConfusion	1.0	U

Image 1-3 Relative Expectations	Value	Dist.
Δ Exposure	0 EV	U
Range of Focus	0 mm	U
Intrinsics.ShutterSpeed	1.0	U
Intrinsics.Aperture	1.0	U
Intrinsics.FilmSpeed	1.0	U
Intrinsics.FocalLength	1.0	U
Intrinsics.DepthOfField	-645 mm	U
Intrinsics.SubjectDistance	0.8 m	U
Intrinsics.CircleOfConfusion	1.0	U

Table 4.13: Derived expression of the pairwise relationship of a set of photographs. Some overlap exists between Images 1 & 2 as seen in the relative range of focus. The relative depth of field in this example is negative because the reference images (the first image of the two) contain less depth of field.

Image 1 Expectations	Value	Dist.
Exposure	[8, 10] EV	U
Image 2 Expectations	**Value**	**U**
Exposure	[9, 11] EV	U
Image 3 Expectations	**Value**	
Exposure	[10, 15] EV	U

Table 4.14: Expression of three images that are part of a high-dynamic-range image. Absolute properties are specified. U = Uniform Distribution

Image 1-2 Relative Expectations	Value	Dist.
Exposure	[-3, 1] EV	U
Image 2-3 Relative Expectations	**Value**	**U**
Exposure	[-6, 1] EV	U
Image 1-3 Relative Expectations	**Value**	
Exposure	[-7, 0] EV	U

Table 4.15: Derived expression of the expected pairwise relationship of a high-dynamic-range image set. Absolute properties are specified. U = Uniform Distribution

4.4.4 Multimodal Medical Imaging

Figure 6.4 demonstrates a T1 weighted magnetic resonance imaging slice which is to be aligned with a brain proton density scan. Within the scope of this example this information is read directly from the medical image metadata. Deformation of less than 10% is expected in the structure of these images based on a visual inspection. Using our category based sensor model defined above in Section 4.2.3, we have expressed this example multimodal image registration problem in Table 4.17.

It should be noted that the medical imaging examples and problems possible under our model are a very limited and trivial subset of the range of possible problem conditions. In order to create a model which also maps the problem of image registration within medical imaging a more sophisticated

Figure 4.10: Input images which contain an unknown exposure variation.

Image 1-2 Relative Expectations	Range / Value	Dist.
Translation.X	[-0.15, 0.15]	+Quad
Translation.Y	[-0.15, 0.15]	+Quad
Skew.X	[-0.15, 0.15]	+Quad
Skew.Y	[-0.15, 0.15]	+Quad
Scale	[0.95, 1.05]	+Quad
Rotation	[-0.05, 0.05]	+Quad

Table 4.16: Derived expression of the *required* pairwise relationship of a high-dynamic-range image set. +Quad = Quadratic Distribution

model must be created.

4.4.5 Super-Resolution Imaging

Super-Resolution images contain nearly identical images which are already almost aligned. Figure 4.12 demonstrates a super-resolution pair. Knowledge of the nature of solution that we desire, and the similarity of images in super-resolution problems allows us to specify the problem conditions directly. Table 4.18 outlines the strict requirements that limit the possible solution space of the aligning transform.

4.5 Summary

Using the new variation centric taxonomy of image registration developed in Chapter 3, we have created a model allowing for the specification of image registration problems. In Appendix A we introduced a formal *definition* of image registration and extend this definition into the applied domain. In

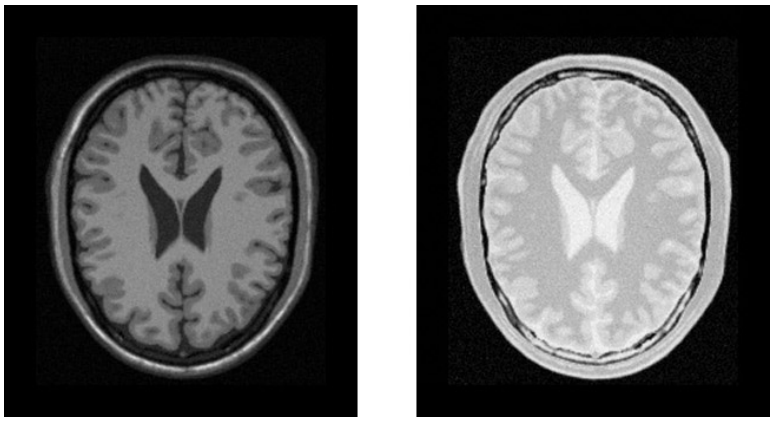

Figure 4.11: A brain T1 slice (L) and a brain proton density slice(R) with possible deformation between images.

Image 1-2 Relative Expectations	Value
Intrinsics.Sensor	T1 / PD
Deformation	[0,10]%

Table 4.17: Expression of the expected pairwise relationship of a multimodal image set.

Section 4.1 we explored the *representation* of the inputs and outputs used in the problem of image registration, allowing developers and researchers to specify the type of solution that they are expecting. Section 4.2 explored the different *conditions* of the problem of image representation, presented as forms of variation. Section 4.3 introduced the necessary concepts and types used in our model, and provided a mapping of the representations and conditions of registration into a formal model through which image registration can be expressed. Finally Section 4.4 demonstrated several common image registration problems under our model. The layers of our framework of accessible computer vision: Problem Definition, Problem Representation, Problem Conditions, and Problem Expression are explored in more detail in Chapter 8.

Our model allows image registration problems to be quantified and classified, providing a framework from which existing algorithms may be investigated. This investigation, which we explore next in Chapter 5, relies upon the framework when mapping the performance of algorithms under different

Figure 4.12: Super resolution input images which contain very little spatial variation, and no other variation.

Image 1-2 Relative Requirement	Range / Value
Overlap	[0.95,1.00]
Δ Ev	0
Focus Value	1.0

Table 4.18: Required pairwise relationship of a super-resolution image set. Absolute properties are specified.

conditions. This mapping further allows us to interpret image registration problem descriptions, selecting the algorithm which performed best under similar conditions.

Chapter 5

A Testbench for Image Registration Algorithms

"The real voyage of discovery consists of not in seeking new landscapes but in having new eyes."

– Marcel Proust

As described in Section 2.1.4, the evaluation of image registration algorithms is currently ad hoc. Researchers use their own images, and often perform a visual analysis of whether the registration 'worked.' Although an attempt has been made [5], no relevant testbench which evaluates image registration across a single sub-problem (such as panoramas) exists, let alone one which covers a range of possible problem spaces and their combinations. The reason for this, in part, is because of the difficulty in creating a testbench which contains ground truth information about the transform which will align the corresponding images. In order to generate a true ground truth transform the images must be synthetically generated, which is a difficult undertaking when taking into account changes in illumination, focus, sensor, and the other forms of variation introduced in Chapter 4.

The testbench outlined within this chapter is not capable of generating perfect synthetic images that exactly mimic these variations for use in interpolated images. Instead it has been designed to create images and ground truth transforms which allow for the analysis of the impact that each type of measured variation has on algorithmic performance. Image pairs are not meant to be directly representative of common image registration problems such as focal stacks or HDR stacks. The testbench contains a significant number of problems which are representative, however no attempt has been made to bias the testbench towards this type of problem. This allows for the mapping of image registration problems which extend beyond the range of

common usage, providing us with a greater understanding of how each form of variation affects performance. Each type of synthetic variation is visually compared directly with images which contain true variation to validate that they are similar.

The range of alignment error of solutions provided by each algorithm across the entire test set is presented to provide a starting point for evaluation. From this basis, the effect that each parameter of the testbench has on the alignment error of an algorithm is measured using a linear regression-based analysis of variance (ANOVA)[38]. ANOVA results consist of two components: a t-value, calculated as the square root of the ratio of the variance between groups to the variance within groups, and a p-value which represents the probability that this variation is due to random factors. For a relationship to be considered statistically significant the p-value must be less than 0.05, and ideally less than 0.01. The magnitude of the t-value indicates the degree to which variation between groups is greater than the variation within groups, with large t-values indicating statistically significant variation.

In addition to the alignment error, the success ratio of the algorithm is significant. Failure can occur in some algorithms if the transform maps the active image completely outside the reference. Similarly, if the feature matching step of feature based methods fail to find enough matches between features, or if the solver step fails to find an acceptable solution, a feature based method may fail. The effect of each test parameter on the success ratio of the algorithm is measured using a logistic regression-based Wald's χ^2 test [79], which calculates the statistical significance of individual regression coefficients. The magnitude of χ^2 similarly indicates the degree to which the test parameter has an impact on the likelihood of belonging to the successful group of results.

This provides an analysis of whether test parameters impact the quality of the solution, the likelihood that a solution will be found by the algorithm, or both. The effect on alignment error is calculated directly using the alignment error of successful solutions. The impact on the success ratio of the algorithm is calculated by dividing the results into two groups based on whether the algorithm provided a solution or not, regardless of the quality of that solution. A one-way ANOVA is calculated for each test parameter, algorithm combination. While this technique does not look for interaction between parameters, insight can still be gained into expected performance based on how significantly each algorithm is impacted by an individual parameter. For each algorithm we present both the t-value and p-value for one-way analysis of variance on each parameter. This is presented

separately for both success ratio and alignment error.

Our testbench tests four algorithms that cover a variety of methods each described within the literature as being useful for solving a different type of registration problem. First, we examine a gradient descent intensity-based method [70] which utilizes the forwards additive method introduced by Lucas and Kanade [53], to solve for the transform, minimizing the square error between aligned image intensities. This was the first method developed for image registration and is described as working reasonably well for image registration problems where intensity remains the same between both images. Second, we test a median-based method developed by Ward et al [100], and modified to work with affine transforms rather than their published translation only transforms, which performs gradient descent on binary maps of the images' median values. This algorithm is described as suitable for exposure varying images. Third, a mutual-information-based method [60] is investigated which also uses gradient descent and is described as being developed to align sensor varying images. While we are not testing sensor variations in our testbench it is possible that this method is capable of solving other types of image registration problems. Finally a feature-based method which utilizes Harris corner detection [33], scale invariant feature transform (SIFT) features [61], and random sample and consensus (RANSAC) [28] to solve for alignment is described as performing well on spatially varying images, and to a lesser extent exposure varying images. These four algorithms were specifically chosen because they should be impacted by different test parameters and have been described in the literature as solving a subset of problems that do not overlap. Although we eventually plan to add more, these methods are sufficient to validate the testbench. Table 5.1 highlights the four algorithms, the settings used in their implementation, and the abbreviations used to represent these algorithms in the graphs and tables below.

The selection of settings is important to the success of an algorithm, and is an area for future research. By evaluating an individual algorithm across the range of image registration problems available in the testbench it would be possible to determine not only the best settings for the algorithm under a given set of conditions, but also the sensitivity of the algorithm to parameter settings. Although initially a computationally intensive and arduous process, this form of testing would only need to be done once, assuming a single testbench which covers the entire problem space of computer vision could be developed. This information could prove invaluable to algorithm developers who often develop an intuition or process surrounding parameter setting, but rarely are able to convey this information in a meaningful way to those who wish to use the algorithm.

Algorithm Details	Settings	Abbreviation
Forwards additive Lukas Kanade gradient descent Sum of square intensity difference error metric	max iterations = 300, search epsilon = 2E-4f	LKSSE
Multiscale regular step gradient descent Sum of square intensity error metric	num levels = 5, Max iterations = 100	ITKSSE
Multiscale regular step gradient descent Binary threshold of median intensity pixels error metric	num levels = 5, Max iterations = 100	MEDIAN
Multiscale regular step gradient descent Mutual information error metric	num levels = 5, Max iterations = 100	MUTUAL
Harris corner detection SIFT Features KD-Tree matching RANSAC Solver	max features = 500 histogram bins = 8, sigma = 1.6 max nearest neighbor checks = 200 forward matching, inlier error tolerance = 0.01	FEATURE

Table 5.1: Details of the algorithms investigated in the testbench.

By examining the results of the testbench we gain an understanding of the performance of these image registration algorithms, and the impact on performance under different problem conditions. The method of examination within this thesis is one dimensional, exploring the effect that each transform parameter or condition has on the alignment error and success ratio. While more sophisticated methods of analysis exist which would allow for the investigation of interactions between parameters, or of particular classes of image registration problems, they are unnecessary for a basic understanding of algorithmic performance under different problem conditions. Understanding these interactions across the entire problem space for each algorithm would require significant effort and would be difficult to explain and demonstrate given the high dimensionality of the space. Instead we extend beyond our basic understanding of performance in the development of our image registration model interpreter which we explore in Chapter 6. This proof of concept interpreter selects algorithms using the mean of testbench results from a specified multidimensional volume within the problem space, ensuring that these higher dimensional interactions are taken into account when selecting an algorithm, even if they are not demonstrated or well understood. A biased testbench which focuses on common classes within the problem space would allow for greater resolution within those regions, but would not operate any differently from our proof of concept.

5.1 Testbench Construction Methodology

Rather than create a fixed set of specific images whose ground truth transforms are known we have created a testbench application which randomly generates known transforms and the corresponding interpolated image pairs. 15K different pairs were created and tested with all algorithms. Algorithms are examined individually, and then a comparison of performance is provided for specific conditions where success ratio and alignment error are significantly impacted.

During the test, parameter data for each transform and image pair is recorded, along with the performance of each of the algorithms, allowing future algorithms to be tested with the same set of transforms and conditions. Images are selected randomly from amongst a set of 40 images which were taken under a variety of conditions, but which are all primarily in focus across the entire image to allow for synthetic focus variation. Figure 5.1 demonstrates a single pair of images, and the corresponding transform which aligns them. These synthetic images are meant to test the effect that

Figure 5.1: Example reference (left) and active (right) image pair from our testbench. The active image corresponds to a transform of: Translation X = -218.87, Translation Y = 29.63, Rotation = -74.27, Scale X = 0.371, Scale Y = 0.557, Skew = 0.179, Overlap = 0.2904.

each form of variation has across the image registration problem space. By testing 15K randomly generated image pairs it is hoped that there will be coverage of a range of problems distributed across the entire volume of the problem space, including both common and uncommon problem conditions.

Table 5.2 presents the range under which each dimension was tested. Limitations in the creation of the transformed image used in this ground truth test limit the range under which some dimensions can be explored. Several of these bounds are based on the limitations of interpolation when rendering images, outlined below in Section 5.1.1. We further limit the testbench to transformations which can be solved by an affine transform.

The starting size of both images is scaled by a factor of 'Image Size' to prevent bias from using images of a particular size. The amount of overlap between the two images is limited by cropping the images from different parts of the source image such that the ratio of overlapping to non overlapping regions in the combined images varies between 0 and 100%. Relative exposure is varied between -3 and +3 exposure values, in 1/3 EV stops, however as with all relative difference properties the absolute value of the difference is used in the ANOVA calculation. Finally, focus varies from 1 to -0.5. Images with a negative focus value are used to represent the alignment of images where no pixels in either image are in perfect focus.

5.1.1 Interpolation Effects

When creating a testbench of synthetic image pairs, special attention must be paid to the interpolation of pixels within the transformed image. With

Parameter	Range	Units
Translation X	[-250,250]	pixels
Translation Y	[-167,167]	pixels
Rotation	[-90,90]	degrees
Scale	[0.25,1]	image size
Skew	[0.0,0.5]	image size
Image Size	[0.25,1]	image size
Overlap	[0.0,100.0]	percent
Exposure	[-3,3]	EV
Focus	[-0.5,1]	focus measure

Table 5.2: Bounds of the testbench parameters. Parameter values were randomly generated, with three configurations of fixed value settings: Exposure = 0 & Focus = 1, Focus = 1, Exposure = 0, to bound the test to particular regions of the problem space.

the exception of whole pixel translations and 90 degree rotations an image which has been transformed will suffer from some degradation and loss of sharpness because pixels in the synthetic image must be interpolated from a weighted average of multiple pixels in the reference image. Figure 5.2 and Equation 5.1 demonstrate how the transformed image's pixels are calculated when using bilinear interpolation while Figure 5.3 provides example images where this pixelation is evident.

$$\begin{aligned} P_t(x,y) &= (1-d)(1-d')P_{1,1} + (d)(1-d')P_{1,2} \\ &\quad +(1-d)(d')P_{2,1} + (d)(d')P_{2,2} \end{aligned} \tag{5.1}$$

In order to limit this type of pixelation we impose two restrictions on our test set. First, in order to ensure that both images are similarly affected we interpolate both the synthetic image and the reference image to roughly the same degree. In order to establish a minimum level of interpolation the reference and synthetic images are scaled to at least 1/2 the resolution of the input image used to generate the test. This results in a similar bicubic interpolation of both images, with both reference and synthetic pixel interpolated from at least nine neighboring pixels. Unfortunately exact matching of the interpolation effect is impossible as the ground truth transform results in a different ratio of reference pixels per synthetic pixel across the image.

The second restriction placed on our test set in order to limit the pixelation effect is a bound on the size of pixels generated by the application of

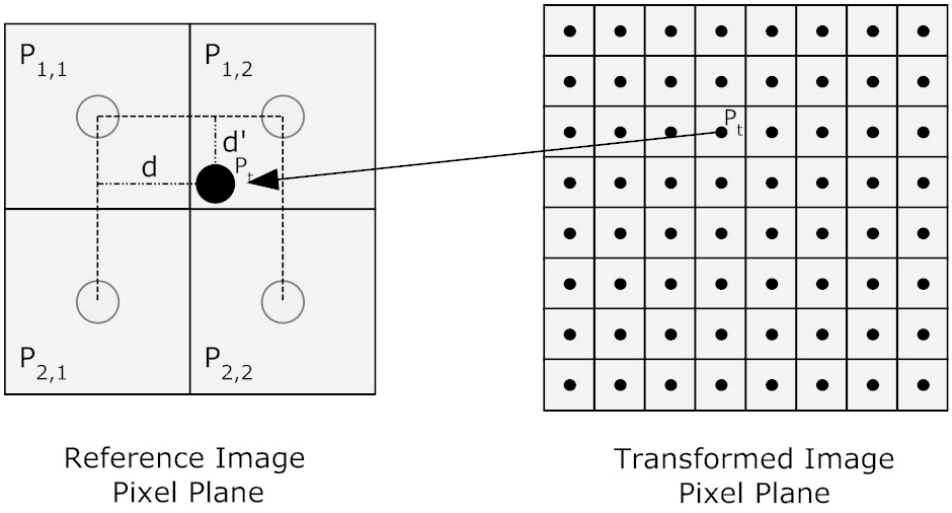

Figure 5.2: Bilinear Interpolation. Transformed pixel value P_t is made up of the weighted sum of the four closest pixels $P_{1,1}$, $P_{1,2}$, $P_{2,1}$, and $P_{2,2}$ of the transformed image.

the transform. Transforms are limited such that the size of a pixel in the synthetic image does not exceed the pixel size of the input image used in the creation of the testbench. Rotation in Z and translation in X and Y do not cause these conditions, however scaling, and to a lesser extent skew and rotations in X or Y (as seen in projective transforms) can cause this type of pixelation. Scale is limited by only scaling the synthetic image such that it is smaller than the reference image. Although this would seem to limit the testbench, the labeling of images once created is arbitrary; by using the synthetic image as the reference in our image registration tests and inverting the ground truth transform we can test images which are scaled both up and down. Skew between images that are part of a panorama, focal stack, hdr image, or multimodal image set is most often limited to within 50% of the image dimensions, and normally found to be within 10%. Skews of this magnitude do not result in pixelation under the conditions previously set up. The difference between X and Y scale follows a similar limitation, and is also limited to within 50% of the image dimensions. Finally rotation in the X or Y-axis, used in projective transforms, can result in transformed pixels which are much closer to the virtual camera, however our testbench focuses on the affine transform. As we will see in our validation of overlap below we also crop the test image to obtain a specific degree of overlap, limiting the

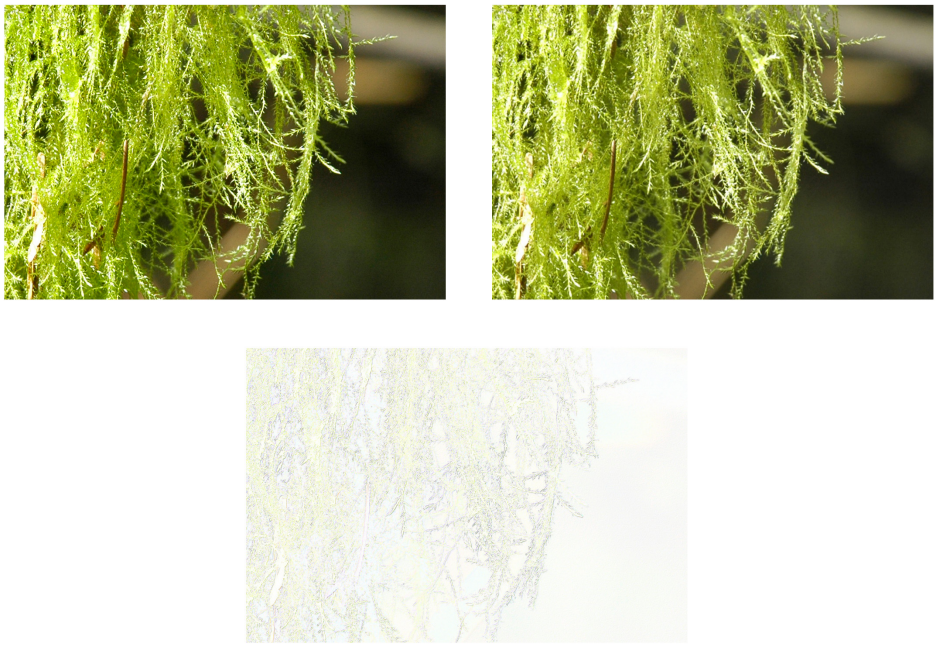

Figure 5.3: Demonstration of pixelation effects due to interpolation. The original image (left) is sharper than the interpolated image (right). Differences between the two images are highlighted in the lower image.

effect that these rotations have.

5.1.2 Synthesizing Exposure Value Variations

As outlined in Section 4.2.1, variations in intensity can be summarized with the concept of exposure values. In order to synthesize variations in exposure that model those expected from an actual change in exposure we utilize the work of Debevec and Milak [23], a seminal work in the creation of high dynamic range images, which calculates the response curve of a camera using a sequence of low dynamic range images, using these camera curves to then create a high dynamic range radiance map of the scene. In our synthesis of exposure variations we instead use these response curves to simulate changes in exposure by calculating the expected value of a pixel based on a particular change in exposure. Although these synthetic images are not exactly equivalent to actual exposure varying photographs, making them unsuitable for the creation of an HDR image, they are similar in appearance particularly

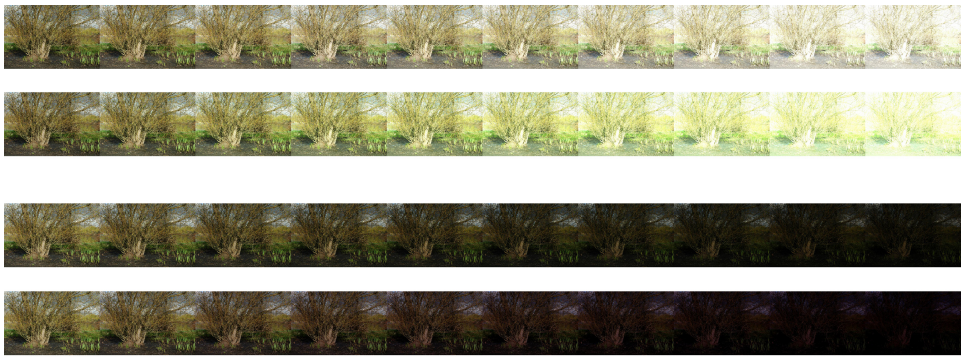

Figure 5.4: Actual variation in exposure in comparison to our synthetic variation. First and third row of images demonstrate actual exposure variations of 0 to +/- 3 in 1/3ev steps. The second and fourth row demonstrate images created using our synthetic image variation process.

with regards to image intensity. Figure 5.4 shows one of the original images which was photographed using a tripod, along with the corresponding actual and synthetic images for exposure variations of 0 to +/- 3 in 1/3 ev steps. From a visual inspection of a range of images our expectation is that the relative performance of image registration algorithms should also be similar.

5.1.3 Synthesizing Focus Variations

In Section 4.2.2 a model of focus was presented, demonstrating the means by which the overlapping in focus regions from an image pair can be calculated. Although depth from focus algorithms which only require a single image [26] would allow us to reconstruct a crude depth map of an image, these depth maps could not be used in the synthesis of images with different depth maps because out of focus regions of the image can not be accurately reconstructed to make them in focus. As an alternative we instead start our synthetic focus variation with an image which is entirely in focus, using an artificial depth map unrelated to image content to create an image which has overlapping regions of focus. Focus is simulated by Gaussian blurring images on a pixel by pixel basis based on the degree to which that given pixel should be out of focus.

Figure 5.5 shows a series of example greyscale depth maps used in order to calculate images with different amounts of overlapping focus. The greyscale values are used to represent a psuedo 'distance' from the camera, or an artificial depth map of the scene. For each image pair the in focus

regions of the image are calculated by multiplying a focus measure which varies from -0.5 to +1.0 by 255 to get a focus offset between -127 and +255. Pixels in focus for the active image are those from 0 to 255 - focusOffset, while those from the reference image are those from focusOffset to 255. Pixels which fall outside this range are out of focus.

In order to simulate the effect of a camera where focus increases exponentially as a function of subject distance we calculate how out of focus a pixel is based on its distance from the nearest in focus range. This is calculated by taking the weighted average of two gaussian blurred images blurred by kernel of size $2^i + 1$ and $2^{i+1} + 1$ where i is calculated from the difference between the pixel and the focus range using the equation $i = floor((dif * 8)^{0.25})$, and weights are calculated as $w2 = (dif * 8)^{0.25} - i$ and $w1 = 1 - w2$. Thus for a focus measure of 0.5, the active image is entirely in focus from depth map pixels of value 0 to 128, while the reference image is in focus from pixels of value 128 to 255. Focus decreases in a gaussian blur such that depth map pixels of value 255 have a difference of 127 for the active image, thus i = 5 and the pixel is constructed from images that are gaussian blurred by a kernel of size $2^5 + 1 = 33$ and $2^6 + 1 = 65$ weighted by $w_1 = 0.354$ and $w_2 = 0.657$.

Under this schema image pairs which vary by a focus of 1.0 are entirely in focus, those that vary by a focus of 0.5 contain a small region of overlapping focus dictated by pixels with a value of 127 in the focus pattern. Images from a pair that has a focus value of 0.0 are only in focus themselves for pixels with a value of 0 or 255. In order to amplify the effect we also explore negative focus values up to -0.5. Image pairs that vary by -0.5 contain no in focus pixels, and each image is blurred by a gaussian kernel of sizes varying between 33 and 257 over the range of the depth map. Figure 5.6 demonstrates an example synthetic focus image pair representative of the images which are used in the testset.

As we saw in Section 4.2.2 focus varies with subject distance and aperture making it extremely difficult to exactly calculate the true amount of focus overlap in an image without a sophisticated setup and equipment. The verification of this synthetic focus variation is therefore left as future work. A visual inspection of the images however shows that they are indeed similar to those used in focus stacks, particularly those where the stack is composed of two images. Comparing the images seen in Figure 5.6 to those in Figure 5.7 we see a distinct similarity. If images with narrower regions of focus such as those seen in multi image focal stacks are required the same method could be modified to create those images by narrowing the range of pixels from the depth map in which the active and reference image are in focus

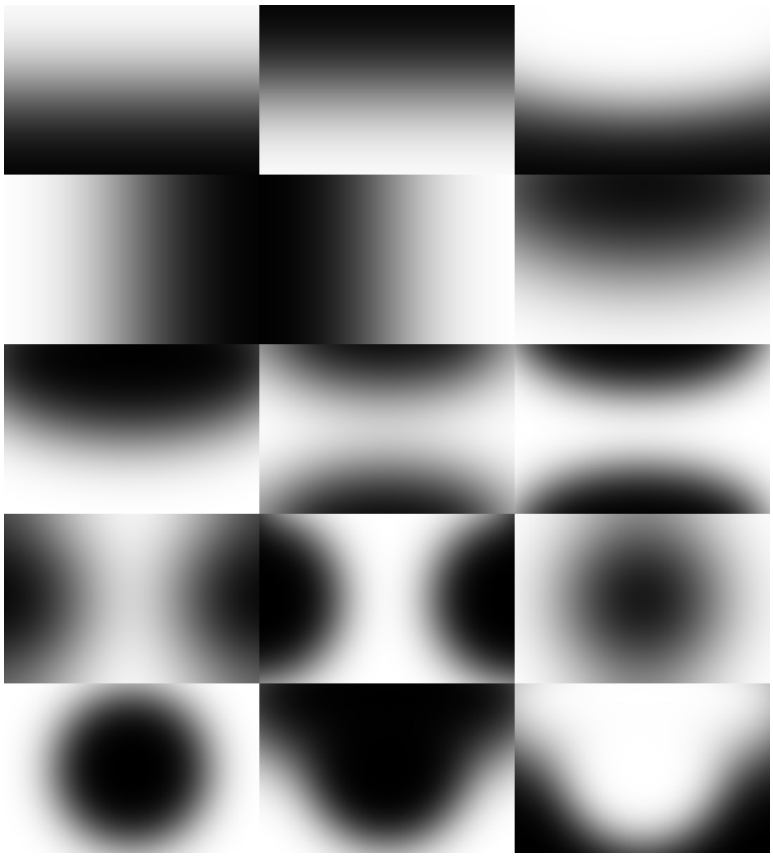

Figure 5.5: Example focus patterns used in the creation of synthetic focus varying images. These patterns simulate distance from the camera, allowing an artificial depth of field to be calculated for the image.

(currently focusOffset to 255 and 0 to 255-focusOffset).

5.1.4 Measuring Algorithm Performance

Testbench error is measured as the alignment error from the resulting transform across the four representative algorithms. Alignment error is measured by projecting a 100 × 100 pixel grid, spaced 10 pixels apart using both the solved transform and the ground truth, and comparing the average difference across pixels between the solutions. This method of analysis is similar to that performed in [5], which also uses known ground truth alignments to solve for the average per pixel alignment error across a grid.

Figure 5.6: Example focus image pair created using our synthetic focus variation. Parameters of the problem are as follows: Translation X = -171.079, Translation Y = 128.034, Rotation = 37.4233, Scale X = 0.839434, Scale Y = 0.799562, Skew = 0.0636235, Image Size = 1, Overlap = 0.957465, and Focus Value = 0.00895718.

In addition to the alignment error, the success ratio of the algorithm, which measures the ratio of solutions found to the total number of tests, is important. Attempts which failed to provide a solution were flagged during the test for further analysis, in order to determine the effect that each test parameter had on the likelihood of success. Failure to find a solution occurs differently depending upon the algorithm being tested. Gradient descent based algorithms are considered to have failed if the resulting transform maps the active image completely outside the reference image. Other measures of success could be developed based on a thresholding of alignment error, however these methods are not a part of the algorithms, which cannot determine whether they have found a correct solution or not. In the case of feature based methods if the feature matching step fails to find enough matches between features, or if the RANSAC step fails to find an acceptable solution, these algorithms report that they have failed, providing meaningful feedback. This measure of success in no way indicates the quality of the alignment found. In order to represent this difference we report alignment error and success separately.

Figure 5.7: Example focus image pair provided for comparison to our synthetic method.

5.2 Testbench Results

Each algorithm was tested with 15K image pairs using a random transform, image, and set of problem parameters in an attempt to provide some coverage across a region of the high dimensional problem space. Three separate sets were created which explore different regions. The first examines 5K image pairs which contain no variation other than overlap, image size, and the aligning transform. In our second test we explore the effect of exposure variations, testing another 5K image pairs which vary in alignment, overlap, image size, and exposure. Finally our third test examines the effect of focus on the available algorithms, testing an additional 5K image pairs which vary in alignment, overlap, image size, and focus. Combining the three of these tests together provides us with a robust mapping of three common regions of the image registration space for a wide range of misalignments, allowing us to measure the impact on performance that each of the test parameters have.

The alignment errors of each algorithm across the entire testbench, ordered from lowest error to highest, are presented in Figure 5.8, giving an idea of the overall expected performance of each algorithm on an unknown problem, i.e. regardless of test parameters. The values of error in Figure 5.8 are surprising for some algorithms, and are discussed in detail in each

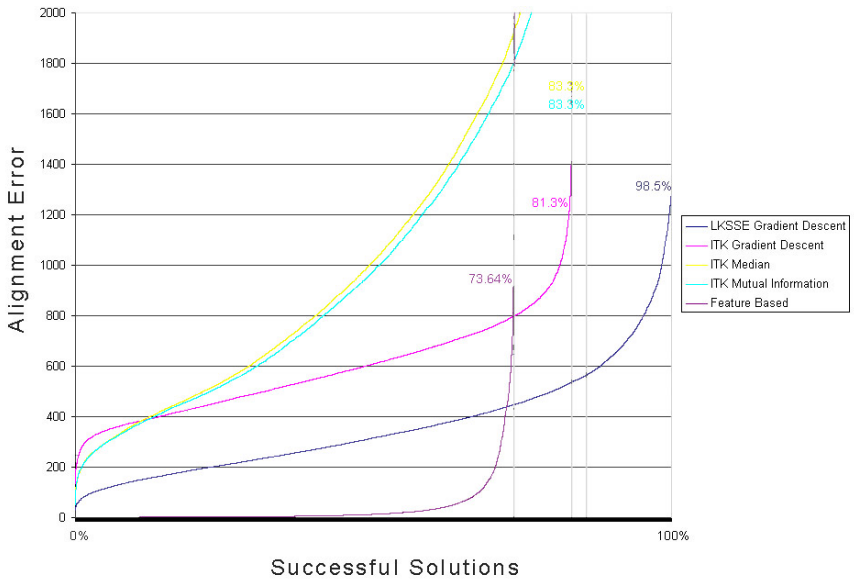

Figure 5.8: The alignment errors of each algorithm across the entire testbench, ordered from lowest error to highest. The success ratio of each algorithm, determined by the number of successful solutions for each algorithm, is highlighted with a dark grey bar.

subsection.

5.2.1 Interpreting Results

As mentioned above, ANOVA results consist of two components: a t-value, calculated as the square root of the ratio of the variance between groups to the variance within groups, and a p-value which represents the probability that this variation is due to random selection. For a relationship to be considered statistically significant the p-value must be less than 0.05, and ideally less than 0.01.

The magnitude of the t-value indicates the degree to which variation between groups is greater than the variation within groups, with large t-values indicating statistically significant variation. The sign of the t-Value indicates the nature of the relationship. Positive t-values indicate that alignment error (or the success ratio) increases as the variable increases, while negative t-values indicate that alignment error (or success ratio) decreases as a function of the variable. This becomes significant when examining pa-

rameters which cause an increase in alignment error in some algorithms, and a decrease in others.

For relative values such as translation in X and Y, rotation, and exposure value, a positive relationship is expected, with greater difference in these values resulting in a higher alignment error. Results for scale in X and Y, and image size are not as straightforward, as they affect the size of the two image pairs, and hence the magnitude of alignment error. The relationship here is likely to be positive with larger images resulting in larger alignment error, unless the algorithm doesn't perform well on small images. Overlap is expected to relate negatively to alignment error, with a decrease in overlap resulting in an increase in alignment error. Finally, the focus value of the image pairs, which varies from -0.5 for the least focused image pairs to 1.0 for the most, could have a different effect depending on the algorithm. It is expected that gradient descent algorithms would see a positive relationship between focus and alignment error, with more in focus images resulting in more alignment error due to the simplification of the search space. Conversely the feature based approach should be negatively related, with a decrease in focus resulting in an increase in alignment error.

The magnitude of χ^2 similarly indicates the degree to which the test parameter has an impact on the likelihood of belonging to the successful group of results. Negative values indicate that as the parameter increases the likelihood decreases, while positive values indicate that the likelihood of success grows as the parameter increases.

5.2.2 Gradient Descent Intensity Based Method

Our sum of square error (SSE) intensity based gradient descent based method uses the forwards additive method developed by Lucas and Kanade [53], to solve for the transform which minimizes the square error between aligned image intensities. According to the literature it should be capable of providing a solution to problems where the intensities are similar.

When initially examining results from the gradient descent intensity based method, some concern was raised as to whether the results were valid. The median alignment error of the algorithm over the course of experimentation was 317.5, with a success ratio of 98.5%. The top 10% of solutions still had an alignment error of 111 pixels; higher than expected. To that end a second implementation of sum of square error gradient descent was also tested using the ITK Toolkit source [68]. In an attempt to improve upon the previous performance a five level multi scale search was used. Reexamining Figure 5.8 demonstrates the alignment error of the ITK im-

plementation ordered by error. As we can see it performs similarly to, but slightly worse than the Lucas Kanade gradient descent implementation with a median alignment error of 557.3, and has a lower success ratio of 83.3%. While the results from the ITK based algorithm are worse, they validate the performance of the other intensity based gradient descent algorithm, indicating that gradient descent based sum of square error algorithms are not especially well suited to image registration problems which differ by a large affine transform. Future investigation into the performance of this algorithm for a small sub-region of the problem space which more closely represents its typical application area could provide insight into the exact limitations of the algorithm.

Examining the impact that each transform parameter has upon the Lucas Kanade algorithm in Table 5.3 reveals that its alignment error is extremely sensitive to rotation, image size, overlap, and exposure, and somewhat sensitive to translation in X and Y, and focus. An increase in alignment error due to image size is logical if incorrect alignments are being found, given that the amount of possible alignment error increases as images get larger. Surprisingly the algorithm positively associates alignment error and overlap, meaning that images with higher overlap contain more alignment error. This could again be due to the fact that they synthesis of overlap results in a larger image, which when misaligned contains higher alignment error. The algorithm's sensitivity to focus is positively related, meaning that the less in focus an image is, the lower the alignment error will be. This makes sense intuitively as out of focus images should be easier for the gradient descent algorithm to search. Its success ratio, presented in the right half of Table 5.3, is similarly affected by changes to overlap and exposure value, and is somewhat affected by changes to translation, scale, and rotation. It is not affected by focus or image size.

Table 5.4 demonstrates the impact that each transform parameter has upon the ITK implementation of gradient descent showing that its alignment error is similarly extremely sensitive to rotation, and image size, and is somewhat sensitive to translation in X, Scale in X, and Overlap. Curiously, it is slightly negatively sensitive to scale in Y, meaning that as scale in Y decreases the alignment error increases. It is not sensitive to exposure, focus, or skew. Its success ratio, demonstrated in the right half of Table 5.4 is extremely affected by changes to image size, followed by changes to overlap and translation. It is only slightly affected by rotation. Success of this algorithm is unaffected by other test parameters.

	Alignment Error		Success Ratio	
Test Parameter	t-Value	p-Value	χ^2	p-Value
Translation X	7.04	0.000	-7.26	0.000
Translation Y	5.59	0.000	-4.95	0.000
Rotation	45.56	0.000	-3.44	0.000
Scale X	3.10	0.002	2.33	0.020
Scale Y	1.77	0.077	5.45	0.000
Skew	0.34	0.734	-1.59	0.112
Image Size	27.82	0.000	-0.60	0.548
Overlap	20.80	0.000	11.37	0.000
Exposure Value	16.90	0.000	-10.18	0.000
Focus	4.24	0.000	-1.79	0.073

Table 5.3: Linear regression-based ANOVA evaluation of alignment error vs. each of the test parameters and logical regression-based Wald test of success ratio vs. each of the test parameters for the sum of square error forwards additive Lucas Kanade based gradient descent method.

	Alignment Error		Success Ratio	
Test Parameter	t-Value	p-Value	χ^2	p-Value
Translation X	17.73	0.000	-10.89	0.000
Translation Y	3.13	0.002	-19.350	0.000
Rotation	29.36	0.000	-3.140	0.002
Scale X	14.02	0.000	-0.720	0.475
Scale Y	-3.96	0.000	-2.270	0.023
Skew	-0.46	0.647	0.62	0.533
Image Size	31.34	0.000	35.620	0.000
Overlap	16.01	0.000	17.810	0.000
Exposure Value	0.61	0.541	2.380	0.017
Focus	-0.33	0.744	-0.490	0.622

Table 5.4: Linear regression-based ANOVA evaluation of alignment error vs. each of the test parameters and logical regression-based Wald test of success ratio vs. each of the test parameters for the ITK gradient descent method.

5.2.3 Gradient Descent Median Based Method

The median based gradient descent approach, modified from the method proposed by Ward et al. [100] to use affine transforms instead of translation only transforms, was designed to align high dynamic range images. Examining this algorithm's plot in Figure 5.8 we see that it performs much worse than the other algorithms, following a similar trend as the ITK gradient descent method, however with a higher slope. With a median alignment error of 807.1, and a mean alignment error of 268.0 for the top 10% of solutions it is questionable whether this method is able to provide a correct alignment for problems with significant affine transforms. The overall success ratio for the algorithm is 81.3% We speculate that the error metric used in this method is not suitable when solving for the six dimensions of the affine transform, explaining why a translation based transform was chosen in their initial implementation of median based gradient descent. Error reporting in [100] was also based on their own image set, which may have been more suitable in terms ranges of overlap.

Table 5.5 demonstrates the impact that each transform parameter has upon Ward's method. Examining the t-values of the graph we see that its alignment error is also extremely sensitive to image size and overlap. The median method is also somewhat sensitive to rotation, and slightly sensitive to translation in X, scale in X, and exposure. It is not sensitive to focus, or skew. The effect on success ratio, seen in the right half of Table 5.5 is most affected by changes to image size, followed by changes to overlap and translation. It is only slightly affected by rotation and exposure value. Success is unaffected by other test parameters.

5.2.4 Mattes Mutual Information Based Method

As with the median based gradient descent method, the Mattes Mutual Information based method, implemented using ITK, was not expected to perform as well as other algorithms. The cyan line in figure 5.8 represents the alignment error of the Mattes Mutual Information based algorithm ordered by error. Performance is quite similar to the median based approach which also used the itk gradient descent method suggesting that without an appropriate error function the search algorithm dominates performance. With a median alignment error of 801.9, and a mean alignment error of 268.9 for the top 10% of solutions it is questionable whether this method is able to provide a correct alignment on image pairs of the type tested. Overall success ratio of the algorithm is 83.3%

	Alignment Error		Success Ratio	
Test Parameter	t-Value	p-Value	χ^2	p-Value
Translation X	-14.89	0.000	-9.58	0.000
Translation Y	-4.48	0.000	-22.02	0.000
Rotation	41.49	0.000	-3.37	0.001
Scale X	-11.73	0.000	1.080	0.282
Scale Y	-2.40	0.016	-1.63	0.104
Skew	0.41	0.685	-0.77	0.442
Image Size	149.47	0.000	37.28	0.000
Overlap	108.18	0.000	19.27	0.000
Exposure Value	6.94	0.000	-4.62	0.000
Focus	1.94	0.053	-1.67	0.095

Table 5.5: Linear regression-based anova evaluation of alignment error vs. each of the test parameters and logical regression-based wald test of success ratio vs. each of the test parameters for the median filtered ITK gradient descent method.

In Table 5.6 we once again see the impact that each transform parameter has, this time upon the ITK implementation of mutual information based gradient descent, showing that its alignment error is extremely sensitive to image size and overlap, which is unsurprising as the amount of possible alignment error increases as these values increase. The median method is also sensitive to rotation, and slightly sensitive to translation in X, scale in X, exposure, and focus. It is not sensitive to skew. The effect on success ratio, again shown in Table 5.6 is most affected by changes to image size, followed by translation in Y, overlap, and translation in X. It is slightly affected by rotation, and is unaffected by other test parameters.

5.2.5 SIFT Feature Based Method

The SIFT feature based method was expected to perform well on the image pairs of the first test set, given that they correspond to the type of images that this algorithm is commonly used to solve. As we see returning to the purple line in Figure 5.8, the plot of the SIFT based method demonstrates its success at this type of problem. The slope of the alignment error is long and flat, although the ratio of success of the algorithm is worse than the Lucas Kanade method. With a median alignment error of 5.44, and a mean alignment error of 1.21 for the top 10% of solutions it is clear that this method is able to provide a correct alignment for a range of problems. The

	Alignment Error		Success Ratio	
Test Parameter	t-Value	p-Value	χ^2	p-Value
Translation X	-15.58	0.000	-10.88	0.000
Translation Y	-4.09	0.000	-19.35	0.000
Rotation	42.22	0.000	-3.14	0.002
Scale X	-10.71	0.000	-0.72	0.473
Scale Y	-3.56	0.000	-2.27	0.023
Skew	0.82	0.411	0.62	0.533
Image Size	152.65	0.000	35.62	0.000
Overlap	111.60	0.000	17.80	0.000
Exposure Value	8.88	0.000	2.38	0.017
Focus	3.29	0.001	-0.49	0.624

Table 5.6: Linear regression-based anova evaluation of alignment error vs. each of the test parameters and logical regression-based wald test of success ratio vs. each of the test parameters for the ITK mutual information metric gradient descent method.

overall success ratio of the algorithm is 73.64% across the entire solution set, which was higher than expected given the range of test parameters.

Examining the impact that each transform parameter has upon the feature based algorithm in Table 5.7 reveals that its alignment error is most sensitive to the amount of overlap between images, with less overlap resulting in more alignment error. It is also sensitive to focus, with less focused images resulting in a higher alignment error. Image size and scale in Y and X, have a similar negative relationship meaning that smaller images contain more alignment error than larger ones. This is likely an indication that there is a lower bound on size of images that this feature based method is capable of solving, which is intuitive given that smaller images will contain less features, and those features will be more difficult to match. Exposure is positively related, with an increase in exposure difference resulting in an increase in alignment error as expected. Other parameters are not statistically significant contributors the changes in alignment error.

Its success ratio, presented in the right half of Table 5.7, is most affected by changes in overlap. It is also significantly affected by focus and image size, and moderately affected by exposure and scale in X and Y. It is somewhat affected by translation, and only slightly affected by rotation and skew.

	Alignment Error		Success Ratio	
Test Parameter	t-Value	p-Value	χ^2	p-Value
Translation X	0.45	0.651	-6.24	0.000
Translation Y	1.78	0.076	-10.32	0.000
Rotation	2.90	0.004	-2.74	0.006
Scale X	-8.40	0.000	13.15	0.000
Scale Y	-10.48	0.000	11.29	0.000
Skew	1.66	0.098	-5.75	0.000
Image Size	-16.94	0.000	29.01	0.000
Overlap	-35.03	0.000	46.74	0.000
Exposure Value	13.22	0.000	-15.42	0.000
Focus	-18.69	0.000	29.61	0.000

Table 5.7: Linear regression-based anova evaluation of alignment error vs. each of the test parameters and logical regression-based wald test of success ratio vs. each of the test parameters for the feature based method.

5.2.6 Summary

Using analysis of variance techniques we are able to measure the effect that each of our testbench parameters has on the alignment error and success ratio of the algorithms tested. The three gradient descent based algorithms did not perform as expected, producing solutions with per pixel alignment errors that were not acceptable for use in most applications. From this we are lead to believe that the use of gradient descent based algorithms for an affine solution space is not tenable. This is supported by the occurrence of translation only, and combined translation and scale transforms within medical imaging and focal stacking applications, where gradient descent based methods are most often used. We have nonetheless evaluated the sensitivities of alignment error and success ratio of these algorithms to each of the testbench parameters, providing an understanding of which conditions are likely to affect the quality of solution on affine problems. A future investigation into the performance of these algorithms for sub-regions of the problem space which more closely represents their typical application area could provide insight into the algorithm's exact limitations.

The feature based method performed better than expected, achieving high quality successful solutions for approximately 2/3 of the dataset. Its alignment error was most affected by overlap, focus, and image size, which matched our intuition about how it would perform. Its success ratio was dramatically affected by exposure value, however it was successful in ˜75%

Test Parameter	LKSSE	ITKSSE	MEDIAN	MUTUAL	FEATURE
Translation X	*	*	*	*	-
Translation Y	*	~	~	~	-
Rotation	!	!	!	!	~
Scale X	~	*	*	*	*
Scale Y	-	~	-	~	*
Skew	-	-	-	-	-
Image Size	!	!	!	!	!
Overlap	!	*	!	!	!
Exposure Value	*	-	~	*	*
Focus	~	-	-	~	!

Table 5.8: Summary of the effect of each parameter on the alignment error of the testbench algorithms, categorized as significant (!), somewhat(*), slight(~) and no effect (-).

of the test cases, 1/3 of which included exposure variation, indicating that it has some success at exposure variations within a particular range.

Table 5.8 summarizes the effect each parameter has on the alignment error of the five testbench algorithms. Table 5.8 summarizes the effect each parameter has on their success ratio.

5.3 Examining Problem Space Dimensions Directly

Section 5.2.1 highlighted the sensitivities of the algorithms tested to the different parameters examined in our testbench. In this section we examine several parameters of interest, dividing the testbench results into groups based on parameter values, and measuring the mean alignment error and success ratio within these groupings to gain a better understanding of algorithmic performance within the problem space. Dividing testbench results based only on the value of a single parameter can conceptually be thought of as taking a one-dimensional slice through the n-dimensional problem space. Although these divisions only examine one dimension, with all other parameters containing any possible value, they still provide insight into the effect that parameter has on the alignment error and success ratio.

Based on the sensitivities of all algorithms, we have investigated the

Success Ratio Test Parameter	LKSSE	ITKSSE	MEDIAN	MUTUAL	FEATURE
Translation X	*	*	*	*	˜
Translation Y	*	!	!	!	˜
Rotation	˜	˜	˜	˜	˜
Scale X	-	-	-	-	*
Scale Y	˜	-	-	-	*
Skew	-	-	-	-	˜
Image Size	-	!	!	!	!
Overlap	*	!	!	!	!
Exposure Value	*	-	˜	-	*
Focus	-	-	-	-	!

Table 5.9: Summary of the effect of each parameter on the success ratio of the testbench algorithms, categorized as significant (!), somewhat(*), slight(˜) and no effect (-).

parameters exposure, overlap, focus, and image size, providing graphs of the mean alignment error for each of the algorithms, and the success ratio of the algorithms, for testbench problems which have been sorted by each parameter respectively.

5.3.1 Exposure Value Variations

The performance of the testbench on the synthetic set of exposure variations can be see in Figure 5.9. As can be seen the mean alignment error of each of the algorithms does change significantly across exposure variations. The Lucas Kanade gradient descent based algorithm and the feature based algorithm have a clear upward trend, with alignment error increasing significantly as exposure variation approaches 3 EV. The median based approach and mutual information based approach both change sporadically, but overall increase slightly as exposure variation increases. Finally, the ITK gradient descent approach is unaffected by exposure value, as indicated in its anova evaluation.

Surprisingly the SIFT feature based method's solutions contain significantly lower alignment error than the other algorithms for problems with 0 to 2 EV exposure variation. The median based approach proposed by Ward et al. [100] which was specifically implemented to solve this type of problem performed much worse than expected, with the worst mean alignment error

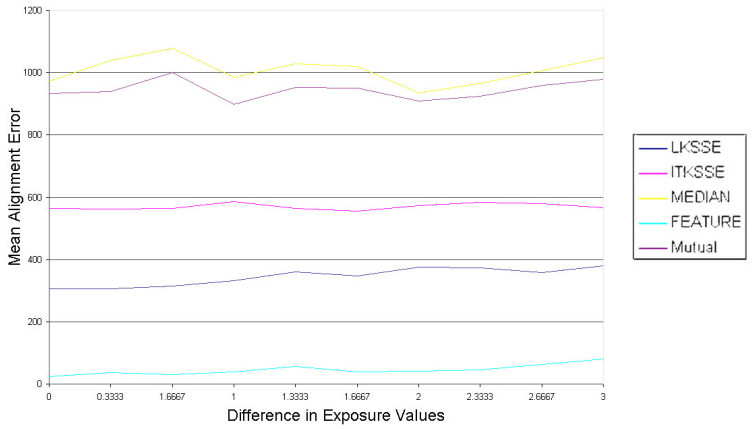

Figure 5.9: Plot of alignment error vs. exposure value across a range of 0 to 3 EV exposure variance. Each of the test results has been sorted into one of ten bins based on exposure value difference, and is graphed separately by algorithm.

of all algorithms. We previously speculated that the gradient descent algorithm used in this method is not capable of solving for the six dimensions of the affine transform, explaining why a translation based transform was chosen in their implementation of median based gradient descent.

The success ratio across the same range of exposure variance is given in Figure 5.10. Examining each algorithm individually we see that the ITK gradient descent and mutual information methods are both unaffected by changes in exposure, while the Lucas Kanade and Median methods' success ratios are slightly affected at extreme changes of exposure. The SIFT feature based method is most affected by changes in exposure, which is unsurprising given that it was not designed with this type of problem in mind and relies on a match between features which are composed of gradients. For exposure increases of ± 1 the algorithm's success only decreases slightly from 85% to 80%. As exposure increases to ± 2 this declines to 63%, and drops even further to 25% at ± 3 EV. Still, for variations of exposure up to 2EV the SIFT feature based method is 63% likely to find a good alignment between the image pair, which was a greater range than we originally thought possible.

Based on this analysis it is clear that for most changes in exposure SIFT feature based algorithms are the best choice for aligning image pairs. When they fail, or when exposure variation is great, the Lucas Kanade gradient descent based method should be used. The median based algorithm de-

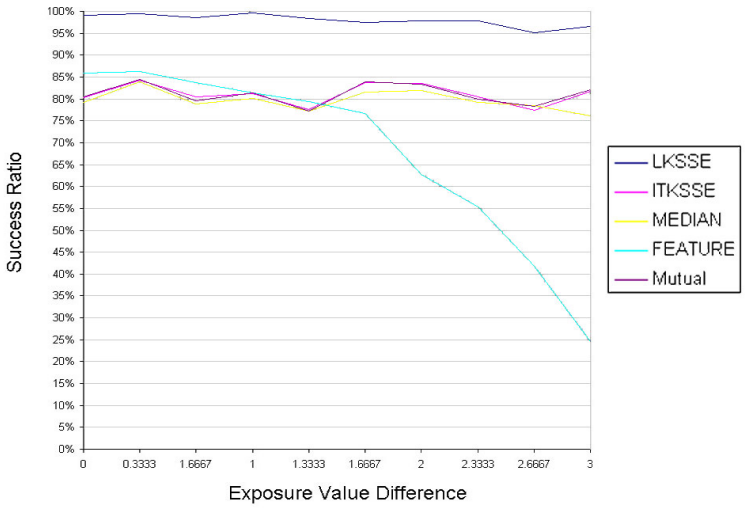

Figure 5.10: Plot of success ratio vs. exposure value across a range of 0 to 3 EV exposure variance. Each of the test results has been sorted into one of ten bins based on exposure value difference, and is graphed separately by algorithm.

signed specifically for high dynamic range images performed much worse than expected and is not likely to be useful in our interpreter.

5.3.2 Overlap

Figure 5.11 graphs the error as a function of overlap for each of the algorithms individually, providing a sense of whether particular algorithms are more sensitive to particular dimensions of the solution space or whether overlap is an algorithm independent concept as is hoped. As expected the SIFT feature based algorithm's mean error is significantly higher for images with 0-4% overlap. As the amount of overlap increases to 24% we begin to see the performance of the algorithm approach the expected range. A spike in mean alignment error exists at the 16-20% range of overlap, most likely due to the corresponding increase in success ratio in that area; the algorithm is finding more solutions, but they're of a mediocre quality. Both of the SSE gradient descent methods are relatively unaffected by overlap, increasing in error slightly as the amount of overlap increases. The median gradient descent and mutual information methods increase significantly as overlap increases; more so than would be expected from the increase in success.

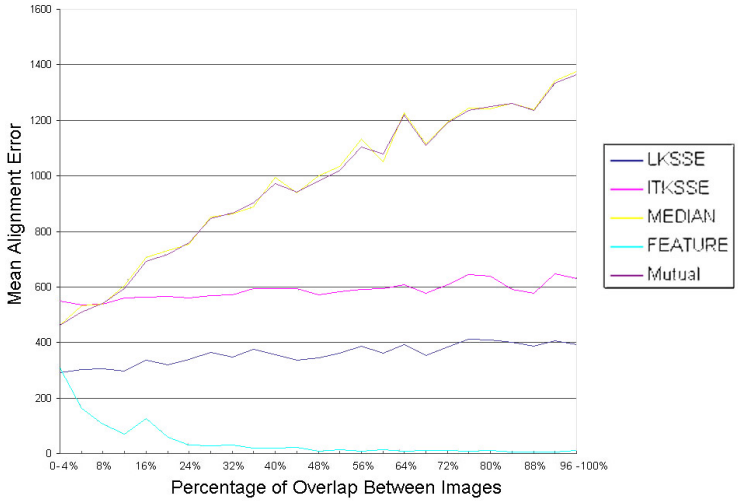

Figure 5.11: Plot of alignment error vs. overlap. Testbench results has been sorted into 25 bins based on overlap value and are graphed separately by algorithm.

Figure 5.12 presents the success ratio of each algorithm as a function of overlap. Most striking here is the effect that overlap has upon the SIFT algorithm. This is expected as fewer matching features are available as the overlap between the two images decreases. At the lowest overlap bin of 0-4% overlap we see that the success ratio of SIFT decreases to 16%. This graph provides an excellent overview of the limitations of the algorithm, suggesting that for overlap between the two images greater than 36% it is 95% successful at finding a solution, however this performance quickly degrades. As expected the other algorithms exhibit a similar decrease in success at low overlap, albeit not to the same degree. In particular the Lucas-Kanade SSE based solver maintains a high success ratio and its mean error is unaffected over the lower range of overlap demonstrating a potential region within the problem space where it should be selected when the SIFT based solver fails.

5.3.3 Focus Variation

The performance of each algorithm on the testbench across the range of possible focus ranges is tested by varying the synthetic focus from -0.5 to + 1.0. Figure 5.13 demonstrates the performance of the testbench on the

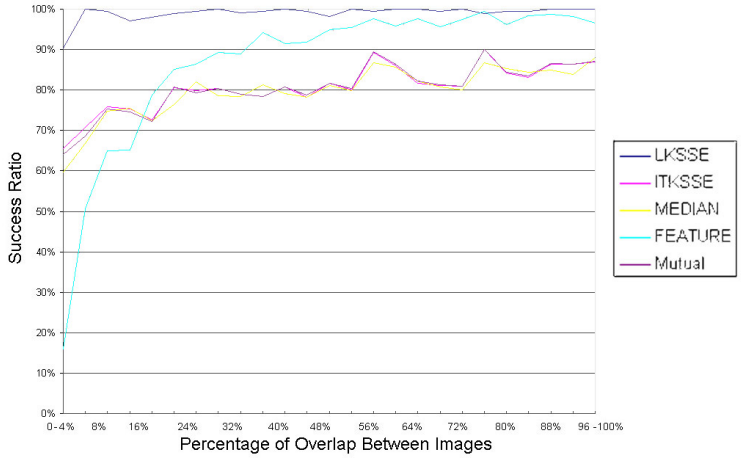

Figure 5.12: Plot of success ratio vs. overlap. Testbench results has been sorted into 25 bins based on overlap value and are graphed separately by algorithm.

synthetic set of exposure variations. The mean alignment error for each of the algorithms across this range of focus variation is plotted. As can be seen from the graph, none of the algorithms' mean alignment errors are affected as focus varies. Once again the SIFT feature based method's solutions contain significantly lower alignment error than the other algorithms, even at negative focus values when neither image is in focus.

Examining the success ratio of the testbench across focus variations we begin to see why. Figure 5.14 plots the success ratio of each tested algorithm as a function of our synthetic focus variation. As expected the Lucas Kanade based algorithm's success ratio is unaffected by focus variation. The three ITK based algorithms, which use sum of square error, median, and mutual information, all report similar success ratios, decreasing slighty as the amount of focus between the images approaches extreme values. Finally the SIFT feature based algorithm is the most impacted, with a 30% success ratio for images with strongly differing focus regions. Surprisingly for image pairs which contain even a slight amount of overlapping in focus regions, shown in our graph as a focus variance of 0.5, SIFT is able to find a solution with a success ratio of 79%. This is significantly higher than we expected, and shows that a SIFT based algorithm may actually prove feasible for solving some focus stacking problems.

Based on this analysis it is clear that for some variations in focus SIFT

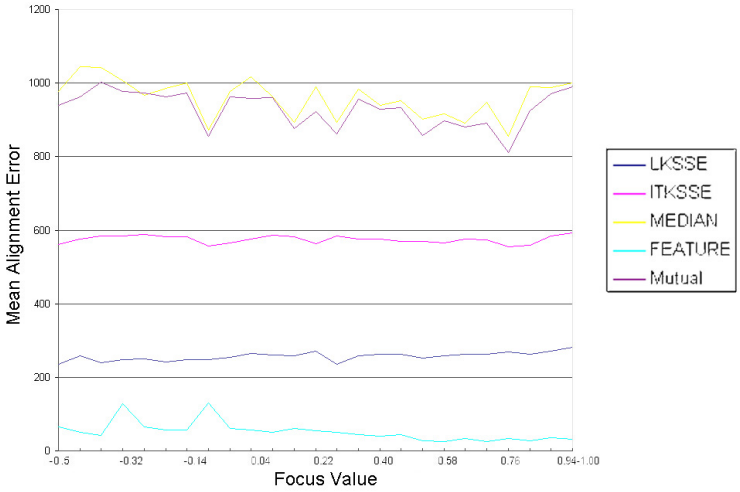

Figure 5.13: Plot of alignment error vs. focus value. Testbench results has been sorted into 25 bins based on overlap value and are graphed separately by algorithm.

feature based algorithms are still the best choice for aligning image pairs. When they fail, or when focus variation is great, our Lucas Kanade gradient descent based method should be used.

5.3.4 Image Size

In order to avoid unknown bias to our system by only using images of a specific size, each test pair is also scaled by a random image size varying between 0.25 and 1.0. Both images are scaled equally before any cropping or image synthesis steps are taken. As we see in Figure 5.15, image size plays little role in the mean error of our Lucas Kanade method, the itk sum of square error method, and the SIFT feature based method. Both the median based and mutual information based methods were, however deeply affected, likely due to the mismatch between their error function and the solution space. The gradient of alignment error as an image is translated, rotated, scaled, or skewed does not correspond to the error function gradient of each of these methods. As images grow in size, the possible error between them increases, and both of these methods are affected, demonstrating the poor quality of solution that these algorithms provide.

Examining the success ratio of each algorithm across the same range of image size in Figure 5.16 we see that the Lucas Kanade method slightly

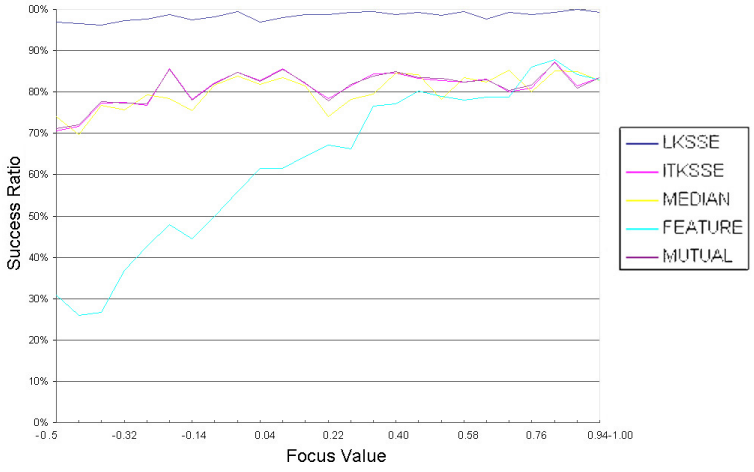

Figure 5.14: Plot of success ratio vs. focus value. Testbench results has been sorted into 25 bins based on overlap value and are graphed separately by algorithm.

decreases in success as the images get larger, while the other four algorithms are significantly impacted, increasing in success ratio as image size grows. At the smallest image size the SIFT feature based method is half as successful as it is for full sized images. The three remaining ITK based methods are even more affected, with a success ratio of only 5% for images of 0.25 image size, explaining the lower mean amongst those algorithms for images of that size. Based on the starting size of the testbench and the range of possible image scaling due to the transform these image pairs would be between 250×375 pixels and 63×94 pixels in size. This knowledge of which algorithms are affected by image size, and what their lower bound on image size is allows for the better selection of algorithms under those conditions as we begin to develop an interpreter.

5.4 Actual Error vs. Reported Error

As a final exploration of the algorithms evaluated by our testbench we examined the actual alignment error in comparison to the error function of the algorithm. SIFT does not measure error directly in this way, and thus was excluded from the process. Figure 5.17 demonstrates the actual alignment error on the x axis of the graph, while the error predicted by the error function of the algorithm returned as part of its error function reporting is

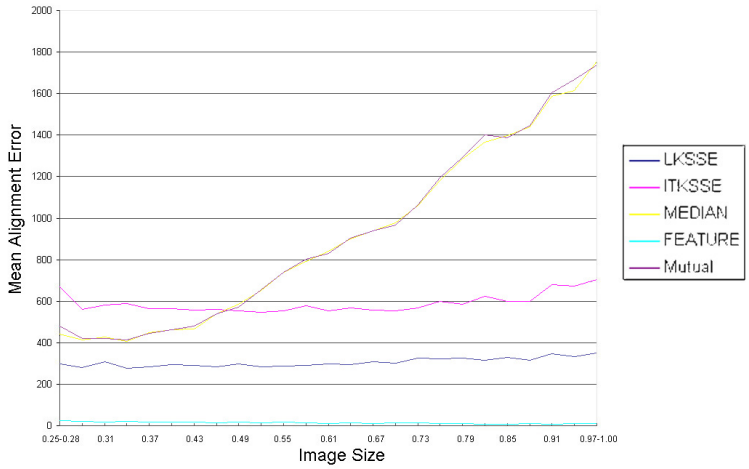

Figure 5.15: Plot of alignment error vs. image size. Testbench results has been sorted into 25 bins based on overlap value and are graphed separately by algorithm.

contrasted on the y axis. As can be seen there is little correlation between actual error and reported error. From this we propose that drawing conclusions about algorithm performance based on these reported error rates is not possible.

5.5 Conclusions

In this chapter a method for creating image registration testbenches was detailed, which uses synthetic variations and a ground truth transform to create image registration pairs with known transform and image parameters. Three testbenches of 5K image pairs were created, each of which covers a range of problems with a common form of variation. No variation, exposure variation, and focus variation regions of the image registration problem space were tested.

Our testbench tests four algorithms that cover a variety of registration methods: a gradient descent intensity-based method [70]; a modification of a median-based method [100] that performs gradient descent on binary maps of the images' median values; a mutual-information-based method [60]; and finally a SIFT feature-based method [61] which uses RANSAC [28] to solve for alignment. Although we eventually plan to add more algorithms, and to test the effect of the modification of algorithm parameters, the method-

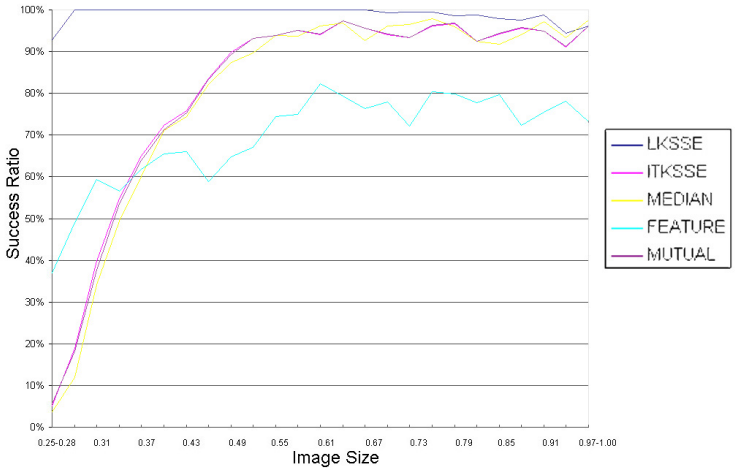

Figure 5.16: Plot of success ratio vs. image size. Testbench results has been sorted into 25 bins based on overlap value and are graphed separately by algorithm.

ology will be identical and these algorithms and settings are sufficient to validate the testbench. Sensitivities of each algorithm to the test parameters were measured using ANOVA, and the impact of several key parameters was explored directly, giving an indication of algorithmic performance along specific problem dimensions.

The performance results of the testbench were somewhat surprising. The gradient descent based algorithms performed poorly outside the limited 'stacking' problems which they are described within the literature as being suitable for. The testbench allows us to examine in detail the effect that each parameter has upon these types of algorithms, revealing an extreme sensitivity to rotation, image size, and the amount of overlap between images. The use of gradient descent for problems with these conditions, particularly if solving for an affine transform are therefore not recommended. Conversely, the range of the image registration problem space the SIFT feature based implementation was able to cover was much greater than expected. As expected it was sensitive to image size, focus, and overlap, however its performance even under extreme instances of these parameters was surprisingly good. It was able to find a good alignment in spite of variations of $\pm$2EV in exposure, almost no focus overlap, and for an image overlap of as little as 0.04. Although its performance breaks down in a number of regions it is much more capable of dealing with variation than expected, and is a good

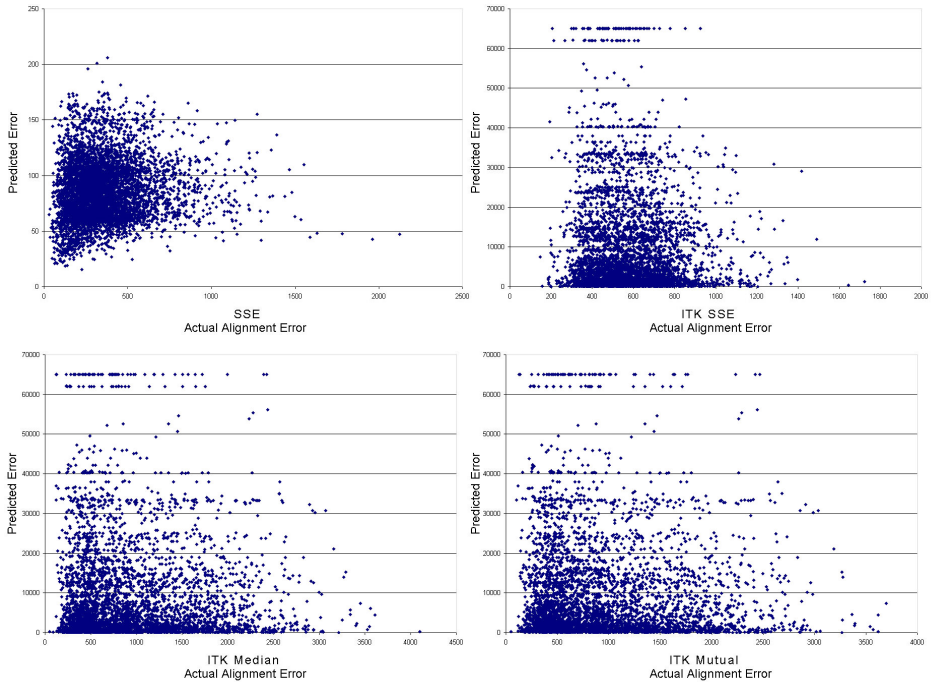

Figure 5.17: Actual alignment error (x axis) vs. Predicted error (y axis) for the four algorithms tested which contained an error function.

candidate algorithm if not much is known a priori about potential problem conditions. The other algorithms tested did not perform as well as expected, possibly due to difficulties of the gradient descent algorithm at solving the six dimensional affine transform. Still, for a number of areas within the image registration problem space the Lucas Kanade intensity based method proved superior.

This testbench provides a basis from which the limitations due to transform parameters, image size, overlap, exposure variation, and focus variation of other algorithms and parameter settings can be evaluated. By mapping the performance of algorithms under these conditions we have created a basis for the selection of appropriate algorithms under similar conditions. A further evaluation of more algorithms, parameter settings, and problem space regions is necessary to create the 'ultimate registration method' described by Zitová and Flusser at the end of their survey, however this evaluation should be straightforward using synthetic variaitons based on the model from Chapter 4, and following the methodology outlined within this chapter.

Chapter 6

Interpretation of the Image Registration Model

"The conventional view serves to protect us from the painful job of thinking."

- John Kenneth Galbraith

In order to validate the registration taxonomy and the model derived from it, interpretability from the problem centric model into appropriate solutions must be shown. There are many possible ways to interpret the description; this chapter demonstrates one approach that utilizes our proof of concept OpenVL machine design. Our interpreter is based on the direct evaluation of the performance of individual registration algorithms under different conditions performed in Chapter 5. From this analysis of how algorithms perform on images which contain similar problem conditions, a single algorithm can be definitively chosen based on which performed best on the test images. In cases where no algorithm was decisively best, multiple algorithms can be selected and executed and their performance compared. Other interpretation techniques are proposed in Section 8.2.2.

The four algorithms introduced in our testbench have been implemented and integrated into our interpreter, providing basic but well rounded image registration capabilities. They are: a gradient descent intensity-based method [70]; a modification of a median-based method [100] that performs gradient descent on binary maps of the images' median values; a mutual-information-based method [60]; and finally a SIFT feature-based method [61] which uses RANSAC [28] to solve for alignment. Although three of the algorithms performed poorly across a wide range of the testbench, they each are targeted towards niches of the problem space which were not necessarily

well represented. The mutual information based method in particular targets a region of the problem space not at all explored, while the gradient descent and median based methods were designed for stacking type problems which represents a small subset of the problem volume explored. They are however still included in the interpreter, which will select these algorithms if the appropriate conditions arise.

Although we eventually plan to add more, these methods are sufficient to validate the frameworks ability to translate expectations and requirements into registration solutions. The integration of new algorithms that are more appropriate under particular situations requires only that they be evaluated with a similar registration testbench, allowing researchers to contribute novel algorithms to the interpretation engine of our model easily. The testbench is made available online as both binary and source in order to encourage the testing of additional registration algorithms. Further, the testbench can be extended to include new types of registration problems, and the included existing algorithms can be evaluated under those new conditions to provide a point of comparison.

6.1 Interpretation of a Model

In order to create an interpreter of the image registration model we require an understanding of algorithmic performance across the different ranges of our model. Our testbench, presented in Section 5.2, provides us with a detailed understanding of how the different forms of variation available in our model affect the performance of algorithms. This allows us to select an appropriate algorithm based on the variation present in the model. There are many ways of using this information to create an interpreter and we examine a variety of methods in Section 8.2. As a basic proof of concept we have implemented a direct, mean alignment error based interpreter which uses the results of registrations from our testbench to select an appropriate algorithm.

When described using the expectations of the image registration model an image registration problem becomes a volume within the problem space. Unspecified representations are assumed to be unknown and to vary across the entire solution space, while unspecified conditions are assumed to be non varying. Using this volume, we determine the mean alignment error, standard deviation, and success ratio of each of the five algorithms for all points in the testbench which fall within it. In instances where a single algorithm's mean error plus its standard deviation is less than the next

Algorithm	Error Space
Intensity-Based Gradient Descent	Sum of square difference of overlapping pixel values
Median-Based Gradient Descent	Sum of square difference of thresholded median pixel values
Maximization of Mutual Information	Sum of square difference of mutual information metric
SIFT Feature Based	Joint intensity of image patches *

Table 6.1: Error metrics of each of the algorithms evaluated by our testbench used to determine which algorithm has performed best when multiple algorithms have arrived at a valid solution.

highest algorithm it is used to solve for the transform. If the algorithm fails to find a solution that meets any requirements on the solution space the next best algorithm is run using the same criteria.

When algorithms are similarly capable of solving a problem, measured by the fact that the mean plus standard deviation of an algorithm is higher than the mean of the others, then a choice must be made in terms of whether to simply run one, or whether to compare the results of all viable algorithms. In order to determine which algorithm has performed best when multiple algorithms have arrived at a valid solution we would require a normalized error metric that applies across all algorithms. While such a metric is not directly available, it can be loosely approximated by cross correlating the results from each algorithm in the error space of the other selected algorithms. For example in the case of two algorithms A and B: we calculate A.Error(Solution A), A.Error(Solution B), B.Error(Solution A), and B.Error(Solution B). These values can also be weighted by the algorithm's estimate of how well they should perform on problems of this type if information on this is available. The metric space of each of our proof of concept algorithms is outlined in Table 6.1. Each algorithm is represented by its representative error space, with the exception of feature based algorithms, whose error space is not possible to evaluate with only a transform. In that case we use the joint intensity of image patches to represent the error space.

While this interpretation method provides us with an estimation of algorithmic performance on the problem for regions of the problem space similar to those covered in our testbench it is unable to estimate for problem areas not covered such as sensor variations or intersections between exposure value and focus. The expansion of the testbench to include new regions of the problem space is an ongoing process. The development of a testbench which covers the intersection of different forms of variation would require significant effort as the number of intersections grows exponentially as new forms of variations are explored, and is left as future work.

Although results vary significantly depending on the volume of problem

space covered by the model's description, we can make some general conclusions about which algorithm will be selected by our interpreter based on the results of our testbench. These inferences are based on our one dimensional evaluations of the image registration problem space, which does not capture the interactions of multiple forms of variation that our interpreter uses, as well as a qualitative analysis of the interpreter.

6.1.1 Purely Spatial Variations

In the case of no variations beyond those found in the transform, SIFT feature based algorithms are most often selected as the best available algorithm, except in the case where overlap varies between 0 and 8%, where the Lucas Kanade gradient descent based method is also run and the error of both solutions is compared. In instances where the SIFT feature based algorithm fails to find a solution the same Lucas Kanade gradient descent based method is run as the secondary algorithm, along with the ITK gradient descent based algorithm, which has a lower mean alignment error than the mean alignment error plus standard deviation of the Lucas Kanade method.

6.1.2 Exposure Variations

When exposure variations are present SIFT feature based results are still often the best choice for finding a solution. If SIFT fails, as is likely to be the case at extreme instances of exposure variation, then both gradient descent based methods are used as the secondary algorithms as was the case of no variation. Ideally another image registration algorithm with less sensitivity to differences in exposure variation such as [87] could be tested and integrated to provide coverage within this region of the problem space, however return to Section 3.2 we see that few other algorithms exist which deal with this form of variation explicitly. It is our hope that the availability of a testbench which measures performance under those conditions will encourage researchers to develop algorithms which are more accurate and more successful within this region of the image registration problem space.

6.1.3 Focus Variations

When exposure variations are present SIFT feature based results are once again the best choice for finding a solution. The accuracy of alignment when using SIFT is greater than that of the other algorithms. If SIFT fails, as is likely to be the case at many instances of focus variation, then both gradient descent based methods are used as the secondary algorithms. This

once again highlights the need for the examination of more algorithms using this testbench. Similar to exposure variations we hope that by making a testbench available we will encourage the development of algorithms which are more accurate and more successful within the focal varying region of the image registration problem space.

6.2 Example Problems

Our examples are intended to illustrate that the various models specified by the expectations and requirements provide a mechanism for the application programmer to specify the problem they want solved. This specification sets the context of the OpenVL state machine. While our demonstration implementation only contains five registration algorithms, the interpretation of the context illustrates that there is enough information for sophisticated algorithmic selection approaches. Our proof of concept shows how the solution to registration problems can be found based on a representative model which describes the vision problem.

The four example problems we examine are: image stitching (panoramas), focal stacking, high dynamic range imaging, and finally a multimodal medical imaging problem.

6.2.1 Stitching

Our first example examines the image stitching problem; finding the alignment between a sequence of images with little spatial overlap. Returning to Section 4.3, Table 6.2 repeated here, outlines the expectations and requirements used to describe our example stitching problem. In this instance the overlap has been specified based on an analysis of image pairs from the problem type. As we saw in Chapter 4 there are many ways of determining the model values including problem analysis, visual analysis, and use of metadata to calculate model values directly.

In our proof-of-concept system, the interpreter selected the SIFT feature-based method to evaluate the transform because its mean alignment error plus standard deviation was lower than the mean alignment error of the other algorithms. For the specified amount of overlap it was decisively the best algorithm of the five available with a mean alignment error of 11.69 plus a standard deviation of 20.20 for a maximum probable error of 31.89 as compared to a mean error 308.974 for intensity-based gradient descent, 1018.25 for median-based gradient descent, and 1004.06 for maximization of

Image 1-2 Relative Expectations	Range / Value	Dist.
Overlap	[0.05, 0.60]	+Quad
Image 1-2 Relative Requirements	**Range**	**Dist.**
Overlap	[0.05, 0.50]	+Quad

Table 6.2: Example panorama stitching registration problem expressed as the relative relationship between a pair of images. +Quad = Positive Quadratic Distribution

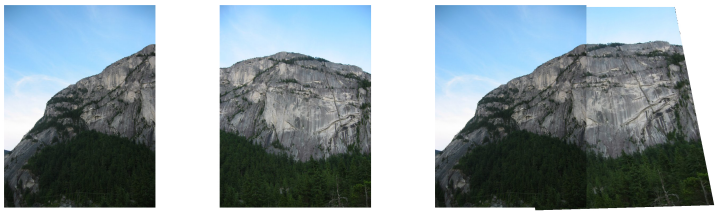

Figure 6.1: Input images and rendered image derived by stitching two images together using a feature based method.

mutual information. Figure 6.1 shows two input images aligned using the feature based method.

6.2.2 Focal Stacking

The focal stacking problem requires transforms from registration of partially in-focus images. Table 6.3 presents an example expectations and requirements based representation. In this case, we have specified the absolute in focus regions for each of the three images, estimating based on our observation of the images. We have also specified that the relative transform that aligns all of the images is of a similar scale, with limited translation and rotation. As we saw above, the expected limited overlapping regions of focus is particularly suited for some registration algorithms while likely problematic for others.

This mapping of focal depth must be converted in the case of our direct interpreter into a mapping similar to that used in the testbench. For the three images which make up the focal stack in this model we can convert the focal depth to similar ranges by using the ratio of coverage of the focal

Image 1 Absolute Expectations	Value	
In Focus Regions	[0.0,0.05]	Uniform
Image 2 Absolute Expectations	Value	
In Focus Regions	[0.05,0.25]	Uniform
Image 3 Absolute Expectations	Value	
In Focus Regions	[0.20,0.50]]	Uniform
Image 1-2, 2-3, 1-3 Relative Expectations	Value	
Translation.X	[-0.25,0.25]	Uniform
Translation.Y	[-0.25,0.25]	Uniform
Scale	[0.95, 1.05]	Gaussian
Rotation	[-0.05,0.05]	Gaussian

Table 6.3: Example state representation for a focal stacking registration problem.

depths to create focus values which match those used in the testbench. The focus value for image 1-2 is $((0.05 + 0.20)/0.25)/2 = 0.5$, as the combination of the two images creates a full coverage, but no regions are overlapping. Focus value for 2-3 is 0.556 ($((0.20 + 0.30)/0.45)/2 = 0.556$), reflecting the overlap between the two images, while the focus value for 1-3 is 0.35 ($((0.05 - 0.30)/0.50)/2 = 0.35$).

The relative and relative-dependent properties specified in Table 6.3 lead to the selection of both the SIFT feature based method as the first choice. For the specified amount of overlap it was decisively the best algorithm of the five available with a mean alignment error of 40.58 + standard deviation of 105.67 for a maximum probable error of 146.25 as compared to a mean error of 257.17 for intensity-based gradient descent, 924.004 for median-based gradient descent, and 1004.06 for maximization of mutual information. However, the SIFT feature based algorithm fails requiring the Lucas Kanade gradient descent based method descent based method be run. Its standard deviation is 96.55, for a maximum probable error of 353.72; less than the mean of the ITK gradient descent based solver. As expected the Lucas Kanade produces an image with the lowest error when compared across the error space of the two algorithms. Figure 6.2 shows the results of applying the transforms aligned by the intensity based solver to a focus-based renderer for a focal stack of six images.

Figure 6.2: Rendered image derived from six images aligned using transforms determined by an image-intensity based algorithm.

Expectation	Range / Value	Distribution / Model
Relative Luminance	[0,0]	Uniform
Relative Luminance	[-2,-2]	Uniform
Translation.X	[-0.05, 0.05]	Uniform
Translation.Y	[-0.05, 0.05]	Uniform
Scale	[0.98, 1.02]	Gaussian
Rotation	[-0.02, 0.02]	Gaussian

Table 6.4: Example state representation for a high dynamic range registration problem.

6.2.3 High Dynamic Range Imaging

Table 6.4 presents the expectations of a high dynamic range problem, where three images of 0, +2, and -2 Ev are to be registered. The values for exposure in this example are derived from the exif information and as such are exact. Based on our knowledge of the images we have also specified that the range of alignments for this problem should cover a very narrow range of likely solutions. This has been specified as a part of the expectations, however the solution space is not limited as these are not requirements.

In this case the variation of illuminance leads once again to the selection of the SIFT feature based method. For the specified amount of overlap it was decisively the best algorithm of the five available with a mean alignment error of 14.36 plus a standard deviation of 57.80 for a maximum probable error of 72.16 as compared to a mean error of 314.41 for intensity-based

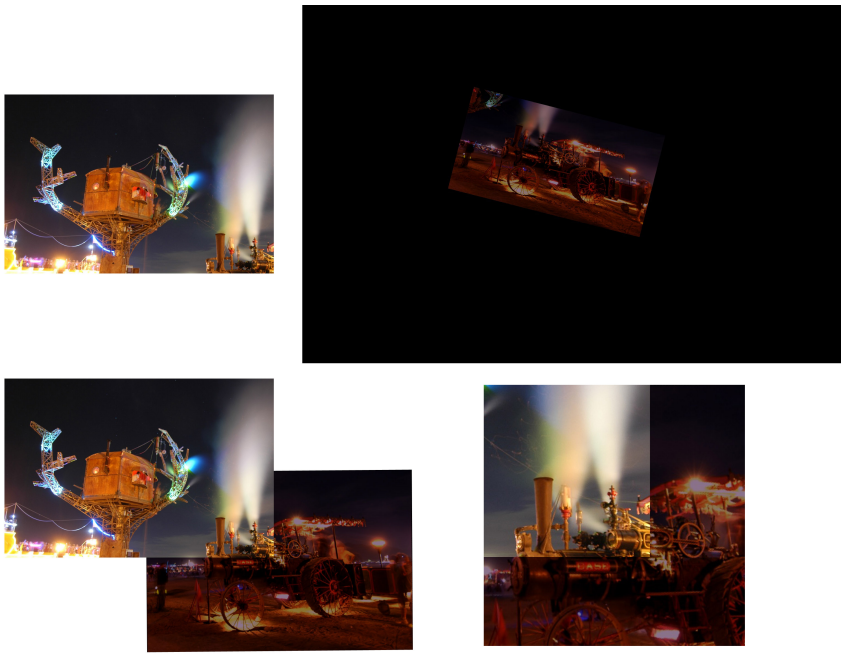

Figure 6.3: Registration of an image pair with a -2.0 exposure variation. Top left is the reference image, and top right the active image. The solution of the SIFT feature based algorithm selected by our interpreter is presented on the bottom. The solution has a mean alignment error of 12.64 pixels/pixel which can be seen in the misalignment of edges where the bottom right hand corner of the reference image merges with the transformed active image.

gradient descent, 1339.8 for median-based gradient descent, and 1203.27 for maximization of mutual information.

In the case of our example the SIFT algorithm found an alignment, eliminating the necessity of running other algorithms. As we saw above in Section 5.2 the SIFT algorithm outperforms the median method which was specifically designed for the problem domain. This helps to illustrate the advantage our language model provides when programmers unfamiliar with the field are implementing registration. Figure 6.3 shows the registration of the image pair with a -2.0 exposure variation by the SIFT feature based method.

Expectation	Range / Value	Distribution / Model
Intrinsics.Sensor	T1	Sensor Model
Intrinsics.Sensor	Proton Density	Sensor Model
Object Model	Head	Single Patient Brain Model

Table 6.5: Example state representation for a medical imaging registration problem.

	Mutual-B-M	Median-B-M	Intensity-B-M
Mutual Error	*0.081*	0.488	0.431
Median Error	0.413	0.292	0.295
Intensity Error	0.440	0.263	0.297
Total Error	*0.935*	1.042	1.023

Table 6.6: Normalized cross-correlation of results for a medical imaging registration problem.

6.2.4 Multimodal Medical Imaging

Our final example examines a medical imaging registration problem where a T1 slice and a proton density slice of a single patient's brain are being combined. Metadata from the medical images allows us to directly specify the model parameters. Although this representation may seem simplistic, the T1 and proton density representations can each be complex models of the actual devices used, allowing researchers to develop algorithms which take this information into account when performing their registration. Complex models of the patient's brain can also be represented, however details on OpenVL's object models are beyond the scope of this thesis.

This problem falls outside of the range of tests performed in our test-bench. In instances where no example problems are available from which to estimate performance we instead implement all algorithms, using the cross correlation of each result in the available error spaces as mentioned above in Section 6.1. Table 6.6 shows the resulting normalized error functions and the cross correlation results for each of the algorithms. As expected, the mutual-information-based method is a clear winner in it's own error space, however in the other two spaces the solution it provides is worse. The

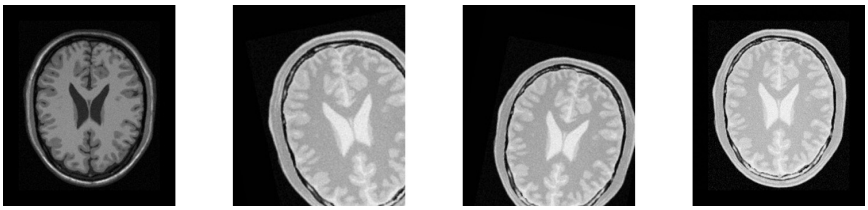

Figure 6.4: Registration of a brain T1 slice to a brain proton density slice. Left most image is the reference image(T1). Solutions presented left to right are: intensity-based, median-based, and mutual-information-based. The feature-based method did not find a solution.

summed cross correlation results indicate that this solution is considered best by the system. Figure 6.4 shows the results of the registration for each of the successful algorithms.

6.3 Summary

Building upon our model of image registration developed in Chapter 4 and our testbench of the image registration problem space from Chapter 5 we have developed a proof of concept interpreter which interprets the model of the problem, selecting the most appropriate algorithm or algorithms to find an alignment.

When described using the image registration model an image registration problem becomes a volume within the image registration space. Our interpreter uses this volume, and determines the mean alignment error, standard deviation, and success ratio of each of the four algorithms for all examples in the testbench which fall within it. In instances where a single algorithm's mean error plus its standard deviation is less than the next highest algorithm it is used to solve for the transform. If the algorithm fails to find a solution the next best algorithm is run using the same criteria. Using this interpreter five image registration problems were explored which cover common areas of the image registration problem space. In each case the interpreter was able to select the most appropriate algorithm, and provide the best solution for the particular conditions of the problem.

The improvement of the interpreter to include new volumes of the problem space or new algorithms relies directly on the testbench methodology developed in Chapter 5. By improving the testbench we see a direct improvement in the interpreter, which is one of the benefits of a direct inter-

pretation method. The incorporation of new algorithms into expert system based interpreters requires a reevaluation of the entire problem space to ensure proper inclusion.

Chapter 7

Automatic Classification of Image Registration Problems

"A computer once beat me at chess, but it was no match for me at kick boxing."

– Emo Philips

Although the field is rapidly moving towards automatic image registration, as we saw in Chapter 2 algorithms and systems are most often limited to a single application domain, such as stitching panoramas [61], super-resolution imaging [29, 104], high-dynamic-range (HDR) imaging [87, 100], or focal stacking [2]. When images vary by more than just alignment the proper selection of appropriate algorithm is critical in calculating the correct spatial transform. From Chapter 5 we determined that techniques can often be used on a range of problems from other domains. In our testbench the feature based method was dominant across a significant volume of the problem space, however it was outperformed in a number of problem space volumes, particularly focus variations. No single algorithm exists that will solve all types of registration. By automatically detecting the important aspects of the image registration model we can determine the volume of the problem space that the image pair falls into, providing a basis for the automatic selection of an appropriate algorithm using our interpreter. Figure 7.1 presents example image pairs from some different types of registration, demonstrating visually the differences between these image pairs.

This chapter introduces two systems which attempt to automatically classify registration problems based on the variation between image pairs identified under the taxonomy of Chapter 3. First, a simple rule based system [72] is explored which validates the idea that the type of registration problem can be identified by features derivable from image pairs. A one to

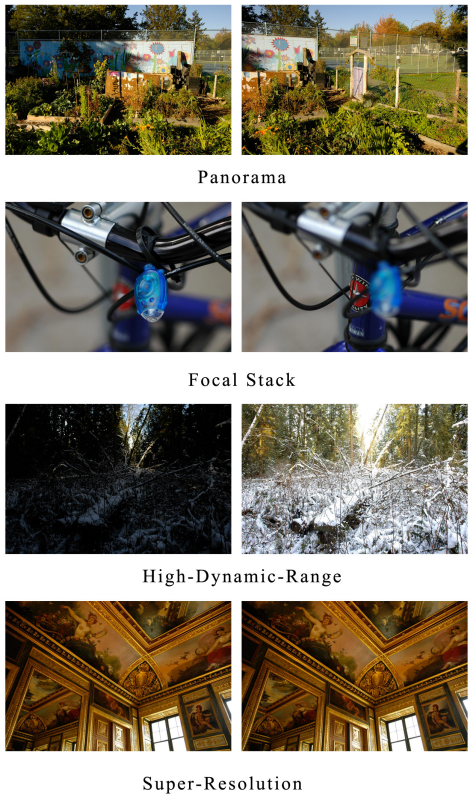

Figure 7.1: Image pairs representative of the different types of registration problems that occur in computational photography. From the top left: panorama, high-dynamic-range, focal stack, and super-resolution.

one classification scheme is used to identify different types of registration problems, and a 90% positive classification rate is achieved for a dataset of 60 images.

A second learning based system developed using support vector machines is then examined in greater detail. In this second study 1100 pairs of images were collected, divided evenly amongst five possible groupings: 220 Panorama pairs, 220 High-Dynamic-Range pairs, 220 Focal pairs, 220 Super-Resolution pairs, and finally 220 'unrelated' pairs, and are made available online to support future research [92]. A one to many classifier was trained which is able to classify between panoramas, high-dynamic-range-images, focal stacks, super-resolution images, and unrelated image pairs with a 91.18%

accuracy. One to one classifiers were also developed to classify each of the categories individually. Classification rates for our one to one classifiers are as follows: Panorama image pairs are classified at 93.15%, high-dynamic-range pairs at 97.56%, focal stack pairs at 95.68%, super-resolution pairs at 99.25%, and finally unrelated image pairs at 95.79%.

Using this classification, a model based representation of the problem can be fed into our interpreter, allowing for the selection of an appropriate algorithm to solve this type of problem. This approach of using the best algorithm available for each problem type is similar to that used in SATZilla [101] which solves propositional satisfiability problems by analyzing the problem to determine the best choice of algorithm for finding a solution. Combining classification with interpretation, our system significantly improves the flexibility and accuracy of automatic registration, providing a starting point for what Zitovã and Flusser [103] term '*the ultimate registration method*,' which is 'able to recognize the type of given task and to decide by itself about the most appropriate solution.'

7.1 Rule-Based Classification

Our first technique for the classification of image registration problems is a crude rule based system, which examines image pairs based on a set of features used to represent the differences and similarities of the pair. Each image pair is analyzed to determine the differences in their intensity histograms and hue/saturation histograms, the normalized power of each image, the number of matching features between the images, and the centroid of those matches. These features are then examined to determine a set of rules which allow us to automatically classify image pairs.

7.1.1 Problem Classification

Image registration methods vary significantly depending on the type of registration being performed. Within our taxonomy image pairs are organized into the categories: non varying, intensity varying, focus varying, and unrelated, based on their primary form of variation. Examining the types of variations that occur in a pair or sequence of images allows photographers to select an appropriate application, or programmers to select an appropriate algorithm, in order to find the best alignment.

Similarly in our rule based system each image pair is analyzed to determine the differences in their intensity histograms and hue/saturation histograms, the normalized power of each image, the number of matching fea-

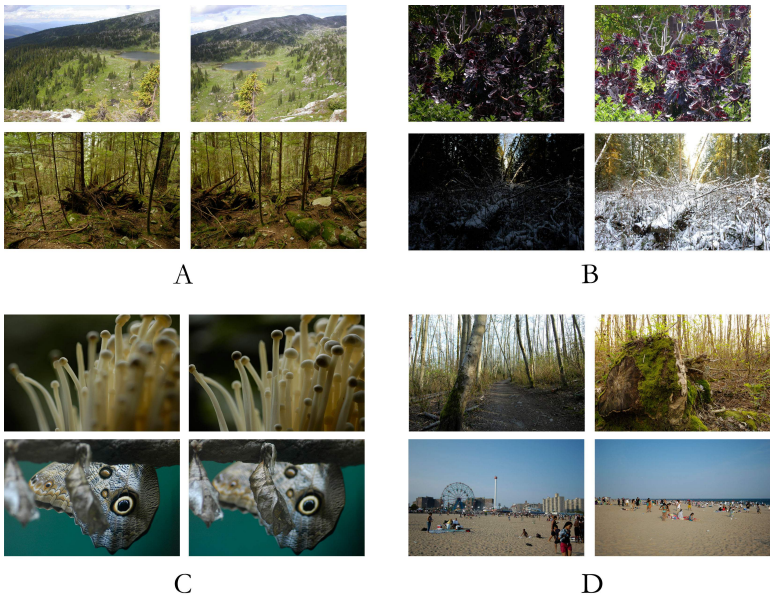

Figure 7.2: Image pairs representative of the different types of variation that occur in registration problems. A: Purely Spatially Varying B: Intensity Varying C: Focus Varying D: Unrelated

tures between the images, and the centroid of those matches. Differences between histograms are measured by their intersection. Figure 7.2 shows two representative image pairs of each type of registration, and Table 7.1 presents their corresponding values. These values are used by the system to classify what type of variations occur through the application of simple heuristic rules that utilize these operators. Our system is capable of running many algorithms and comparing the results to find the best solution, therefore it is much more important to make true positive classifications than it is to prevent false positives. The basis for these rules within each application domain is examined in detail below.

Purely Spatial Variations

Image pairs that differ purely spatially, as shown in Figure 7.2A, are the most common type of image registration problem. Applications that require registration of images that vary spatially include panorama stitching, super resolution, and remote sensing. Although area-based methods derivative of Lucas and Kanade [53] are capable of solving these types of registration

Image Pair	N Feat	Centroid	I	HS	Power
A1 (Spatial)	967	(0.76,0.51)(0.22,0.61)	0.765	0.787	3.30%, 3.10%
A2 (Spatial)	605	(0.71,0.65)(0.30,0.65)	0.923	0.878	3.30%, 3.05%
B1 (Intensity)	944	(0.53,0.62)(0.53,0.62)	0.161	0.303	3.79%, 1.80%
B2 (Intensity)	1483	(0.53,0.60)(0.43,0.50)	0.425	0.651	3.60%, 3.24%
C1 (Focus)	139	(0.52,0.53)(0.52,0.53)	0.834	0.812	1.40%, 0.96%
C2 (Focus)	50	(0.52,0.36)(0.58,0.32)	0.862	0.834	0.16%, 0.43%
D1 (Unrelated)	24	(0.49,0.25)(0.44,0.28)	0.898	0.819	3.18%, 2.76%
D2 (Unrelated)	10	(0.47,0.6)(0.58,0.73)	0.746	0.704	1.88%, 1.73%

Table 7.1: Values used in the classification of image pairs corresponding to images from Figure 7.2. For each pair the number of features (N Feat), feature centroid (Centroid), overlap of intensity histogram (I), overlap of hue saturation histogram (HS), and power of each image (Power) is calculated.

problems, feature based methods like Autostitch [61] and Autopano are the most common technique applied and are generally considered much more accurate unless the image pairs contain little high-frequency information from which to find and match features.

Without first aligning the images, calculating the amount of *overlapping* high frequency content in the image pairs is difficult, so instead we calculate the number of matched features [52] directly. Image pairs with on average more than one matched feature per 75×75 pixel patch are classified as 'purely spatial' because methods unconcerned with other forms of variation (i.e. feature based methods) are likely to be capable of solving for their alignment. Stitching problems with low overlap are likely to contain a low number of matching features, so we also calculate the centroid of the features detected, allowing us to distinguish these cases. Pairs with feature centroids greater than 30% translation from the origin are considered purely spatial require 1/5 as many matches. Section 7.1.3 shows how the results of the combination of these two rules allows us to positively classify purely spatially varying image pairs within our test set.

From this set of rules it is also possible to create a representative model of the area of the image registration problem space that they cover. In this particular case our model is similar to the example presented above in Section 6.2. This model can be used as an input to the interpreter, allowing for not just automatic classification, but also automatic implementation of image registration.

Expectation	Range / Value	Distribution / Model
Relative Luminance	1 - 3	Uniform

Table 7.2: Our model of image registration corresponding to the subset of problems selected by our rule for detection of intensity variations.

Intensity Variations

Significant intensity variations are common amongst high dynamic range image registration problems, and can also appear in panorama image pairs where there is a powerful light source in one of the frames. HDR techniques are predominantly area based; interest points required by feature based methods are most often detected at edges or corners, and are not consistent across large differences in intensity. For those image pairs where image intensity varies significantly, such as those shown in Figure 7.2B, median thresholding [100] can be used to find a more accurate registration.

Intensity varying image pairs can be easily detected by examining the differences in intensity histograms, providing a simple basis for their classification. Pairs with histograms that differ by more than 30% are classified as intensity varying. Section 7.1.3 demonstrates the effectiveness of this rule at finding intensity varying image pairs within our test set.

A mapping of model which is representative of this rule can also be constructed. Table 7.2 presents the model of image registration problems that correspond to this subset of the problem space.

Focus Variations

Focus variations are found in image pairs used for focus stacking, and in pairs with motion or gaussian blur, shown above in Figure 7.2C,. According to the literature, techniques are predominantly area based for the same reason as HDR techniques; the same edges and corners are not detected across images with different focal planes. Instead intensity based area methods like those derivative of Lucas and Kanade [53] are recommended to find the correct alignment. In our testbench we found this to be true for instances of extreme focus variation and have chosen to utilize this method in our algorithm selection, however the feature based method explored in our testbench may be a more appropriate choice if the expected focus variation in the dataset is small.

Focus stacking is used to combine images with limited depth of field,

Image 1-2 Relative Expectations	Value	Dist.
Δ Exposure	0 EV	U
Δ Range of Focus	0 mm	U

Table 7.3: Model representative of image pairs selected by our focus variation detection rule.

so images are likely to have a low amount of high frequency information. Image pairs are detected by examining the normalized power of each of the images, a measure proportional to the number of in focus pixels in the image. In a number of problems, particularly those relating to registering blurred images, only one of the images is lacking in focus. Image pairs where either image has a normalized power less the 2.5% are classified as focus varying. As we will see in Section 7.1.3 this rule is useful for positively classifying focus varying pairs, however it also classifies a number of other pairs which are not considered as primarily focus varying in our ground truth.

A similar model to those presented above can be created based on the rule used to detect focus variations. Table 7.3 presents the model of image registration problems that correspond to the focus varying subset of the problem space.

7.1.2 One to Many Classification System

Using the rules described in Section 7.1.1 our system is able to identify the types of variation occurring between the image pair. Each form of variation is represented within our interpreter as a volume of the problem space expressed as a model. If only a single type of variation is identified then the corresponding model is used by the interpreter to solve for the transform that aligns the pair. When multiple forms of variation are classified for an image pair the system uses a set of volumes to determine the most appropriate algorithm or algorithms. Our proof of concept interpreter then uses one or more of the four algorithms, solving for the best alignment as described in Section 8.2. If multiple algorithms are selected then normalized cross correlation of the proposed transforms is then performed, calculating the error of each transform across all appropriate error spaces to pick the best.

7.1.3 Evaluation

To test our classification algorithm we created a set of 64 image pairs from the categories: spatially varying, intensity varying, focus varying, and unrelated, based on their primary form of variation. These images were then classified in a user study by six independent photographers. For each pair we considered the classification to be valid if five of the six photographers classified the image pairs exactly the same, a process which eliminated four pairs. This set of classified pairs was then used as a ground truth for evaluating the system. For the remaining 60 images the photographers were on average 96% successful at correctly classifying the main form of variation using a one to one classification scheme. This allows us to compare how well our system is able to classify registration problems.

Using our two rules outlined in 7.1.1 we can positively identify 100% of the purely spatial varying problems within the data set. 38% of pairs classified as primarily spatially varying were also proposed as being intensity or focus varying. Once normalized cross correlation has been applied 76% of the purely spatial (according to our ground truth) pairs find the best alignment using the correspond spatial method. Examination of the remaining 24% of spatial pairs shows that in 60% of cases all error spaces agreed the solution chosen was the best, while 40% produced conflicting recommendations.

As expected, intensity varying image pairs can be easily detected by examining the differences in intensity histograms as proposed in 7.1.1. Using this rule we are able to find 100% of the intensity varying image pairs within the data set. 91% of ground truth intensity varying pairs were also indicated by either the spatial and/or focus varying rules. After NCC however 81% of the selected solutions were from the intensity varying method. The remaining 19% were selected from the spatially varying method.

Similarly, using the rule proposed in 7.1.1 we are able to classify 94% of the focus varying problems in our test set. 16% of ground truth pairs were also classified as spatially varying, however after NCC all of the solutions were selected from the focus varying method.

Unclassified image pairs are considered to be unrelated by the system. 38% of the unrelated image pairs were correctly identified by the system. A single focus varying problem was also indicated as being unrelated. This poor rate of classification of unrelated image pairs derives from nature of our rules, which were chosen to identify differences in intensity and focus between images, a common occurrence in unrelated images.

Overall the system is able to positively classify 90% of the registration

Figure 7.3: Image pairs that were aligned using a method other than that suggested by their main form of variation.

problems correctly. Removing unrelated image pairs from the set increases the correct classification rate to 96%. 55% of the problems were correctly classified with no alternative variation suggested and were solved using their appropriate method. A further 32% selected the solution by the corresponding method for their classification through normalized cross correlation. Finally, for the remaining 13% of image pairs, 71% of the solutions selected by the system were lowest in all error spaces being considered, suggesting that they represent a better alignment than that proposed by the 'correct' method. Figure 7.3 shows this set of images. Table 7.4 summarizes the results of our system's performance classifying the test set.

Ground Truth Classification	Identified	W No Alt	After NCC
Spatial	100%	62%	76%
Intensity	100%	9%	81%
Focus	94%	78%	94%
Unrelated	38%	38%	38%
Total Related	98%	56%	87%

Table 7.4: Summary of the system's classification rate.

7.2 Learning-Based Classification

Having validated that it is possible to classify image registration pairs purely based on features extracted from the images, we now explore the concept of classification in more depth using a more sophisticated learning based approach. In our second classification method support vector machines (SVMs) are used to classify image pairs, determining whether a given pair is from a panorama, a high-dynamic-range image, a focal stack, a super-resolution set, or is unrelated. Using this approach we can determine the local volume of the image registration problem space that the image pair is likely to be contained within, making it possible to select an appropriate method of solution using our interpreter. This approach is similar to that used in SATZilla [101] which solves propositional satisfiability problems by analyzing the problem to determine the best choice of algorithm for finding a solution.

In order to classify amongst the different categories of registration application using SVMs a feature vector is needed which describes the image pairs. In the case of image registration the variation between the images is one of the most distinguishing features between types. We have evaluated a wide range of features for this description and present them below in Section 7.2.1. Many of the features examined are general representations of the image as a whole, in the form of histograms or differences of histograms. These representational features are less likely to be affected by the size of the images and as such we have examined the effect of the size of images used to calculate the feature vector on the classification rate of the system. In addition we look at feature importance, evaluating which features are best at distinguishing between the different classes of registration.

Support vector machines are a learning based approach and classify based on a set of training data. A set of 1100 image pairs were created, manually classified according to the type of registration, and used to train and test the system. This set was used to train both one to many classification of the registration problem domain, as well as one to one classification of individual registration problem types, i.e. is this pair likely from a panorama? This one to one classification is useful for validating the performance of the one to many classifier which should see similar results. The set of image pairs used to train the system is made available online [92] for researchers who wish to improve upon these results.

7.2.1 Classification of Image Registration Problems Using Support Vector Machines

Support Vector Machines (SVMs) are a supervised learning technique similar to neural networks, which can be used for classification. SVMs work by projecting a description of the element to be classified, known as a feature vector, into a higher dimensional data space where the different classes become linearly separable. LibSVM [17] and the accompanying scripts were used to find the appropriate SVM parameters for our problem, normalize the feature vectors, train the classification models, test the models, cross validate the results, and explore the relevance of features from our feature vector. We utilize the recommended Radial Basis Function (RBF) based kernel, and linearly scale our feature vector data to [-1,1]. A grid search of the two RBF parameters C and γ was performed to find the best settings for our problem independently for each classifier. To prevent over-fitting a five-fold cross-validation scheme was utilized. This method breaks the test set into five equal subsets, testing each subset on a classifier trained from the other four subsets.

The development of a feature vector that appropriately describes the different aspects of the image pairs to be classified is a critical part of the system. In order to classify image registration we have developed a feature vector to describe both the images themselves, as well as the variation between the image pairs.

Additionally, to train and test the system a significant dataset of image pairs is needed. We have created a set consisting of 1100 image pairs, representing the five classifications of: panorama, high-dynamic-range image, focal stack, super-resolution image, and finally unrelated images. The set is made available to other researchers on our website [92] in order to allow for direct comparison when improving upon our classification results.

Feature Vector

The description of the image pairs by the feature vector is critical in the system's ability to classify. In order to ensure that the SVM has enough information about the pair to correctly classify it we include information about each image, measures of the global relationships between aspects of the image pair, and also local relationship (pixel-wise) between these aspects. In Section 7.2.2 we examine the importance of each feature in our feature vector.

This feature vector is an extension of the one proposed in Section 7.1. For

each image pair in our rule based system we calculate the average intensity, hue, and saturation of the image. The power of each image, which serves as a measure of how in-focus the image is, is calculated by applying a 5×5 Laplace filter to detect edges, squaring the result, and normalizing across the entire image. Finally the number of SIFT features in each image is calculated.

As we saw in Section 7.1 these differences between images provides us with significant insight regarding which category a pair belongs to. Additional information could also prove useful in determining what type of image registration problem the pair relates to. Rather than simply take the difference between average values we compare the overlap of the images' histograms, both globally and locally. Intensity and Power histograms are calculated using 64 bins, and their overlap is calculated as the of intersection between the two dense histograms. Our joint two dimensional Hue / Saturation histogram has 30×32 bins and is compared similarly. The number of matched SIFT features, as well as the number matched per pixel is calculated for the vector. In [61], Brown uses the number of matched features as a basis for selecting the next image to combine into a panorama, performing a similar classification, motivating this inclusion. In addition the centroid of the matched features is calculated for each image.

Finally, to make local image comparisons we divide the images into nine equal regions, comparing the overlap of local intensity histograms for each section of the image. Table 7.5 demonstrates the complete feature vector, and provides example values for a pair of panorama images and a pair of high-dynamic-range images.

Data Set

To train and test the SVM a set of image pairs which are representative of each of our classes under various conditions possible is necessary. In our creation of this test set we have attempted to include image pairs taken in different lighting conditions and settings so as not to unwittingly bias the learning system. Photographs were taken in pairs, with a specific single application in mind, and categorized accordingly. Images were scaled from 3008×2000 pixels to our base size of 1504×1000 pixels in size in order to accommodate memory limitations of our SIFT feature implementation. As we will see in Section 7.2.2, which looks at the feature vector's invariance to scale, this is unlikely to affect performance of the classification of full sized images. 1100 pairs of images were taken in total, divided evenly amongst the five possible groupings: 220 panorama pairs, 220 high-dynamic-range pairs,

220 focal pairs, 220 super-resolution [10] pairs, and finally 220 'unrelated' pairs. Unrelated pairs consist of images taken from within the same category, for example both images are from a focal stack, and are distributed evenly amongst the four application based categories.

The same set of image pairs was used both in the training of our one to many classifier, which labels across all classes, and our one to one classifiers, which attempt to classify whether an image is a part of a given class or not using all other classes as negative training cases.

7.2.2 Evaluation

As mentioned in Section 7.2.1 we have trained and tested our SVM using five-fold cross validation in order to prevent overfitting of our data set. The one to many classification rate of the system for full sized images is 91.18%. This rate of classification makes the automation of image registration tools feasible, and would allow photographers to reliably group sets of photos automatically by type and apply an appropriate registration algorithm.

The one to one classifiers were similarly trained and provide a point of comparison to the one to many classifier. Classification rates for our one to one classifiers are as follows: Panorama image pairs are classified at 93.15%, high-dynamic-range pairs at 97.56%, focal stack pairs at 95.68%, super-resolution pairs at 99.25%, and finally unrelated image pairs at 95.79%. Table 7.6 summarizes these classification rates.

Feature Importance

Examining the importance of the individual features with regards to classification provides insight into how the classifications are taking place. Chen et al. [18] developed a measure of feature importance known as an 'FScore' which measures the discrimination of two sets of real numbers. Table 7.7 presents the FScores of our feature vector for our one to many classifier.

Using the FScore ranking as a basis, features with low FScores can be removed from the system and the impact on classification measured, providing a basis for improvement of the speed of the system at a cost of accuracy. With our complete feature vector we achieved a classification rate of 91.18%. Reducing the number of features to: 30 results in a 90.90% classification rate, 15 results in a 90.52% classification rate, 7 results in 86.59% accuracy, and finally using only the top 3 features results in 85.83% classification rate. Figure 7.4 summarizes these results.

Also of interest is the top three features from each one to one classifier.

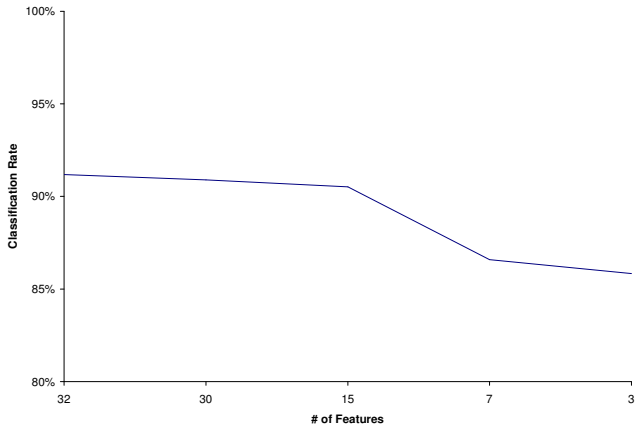

Figure 7.4: Summary of classification rates based on reducing the number of features in the feature vector. Classification rates of feature vectors of the 32, 30, 15, 7, and 3 highest F-Score [18] features are shown.

These provide insight into how classification of each type of registration is being performed. The top three features from each category are summarized in Table 7.8. As expected the most important features in each class relate directly to the common forms of variation that are indicative of that class.

Invariance to Scale

Computation time of the feature vector is exponentially (n^2) related to the size of the images from which it is calculated. Many of the elements in our feature vector are global calculations such as histograms or averages which are unlikely to be significantly affected by the size of the image. As such, we have investigated the impact of image size on the classification rate of our system. To test this feature vectors were generated from 1504 $\times$ 1000 pixel images calculated at: 100%, 80%, 60%, 50%, 40%, 30%, 25%, 20%, 15%, 10%, 5%, 4%, 3%, and 2% scale, and a one to many classifier was trained using 5-fold cross validation. As we see in Figure 7.5 classification remains level at ˜91% until the image is scaled down to 10% of its original size (150 x 100 pixels). Decreasing the size of the images to 2% of their original size

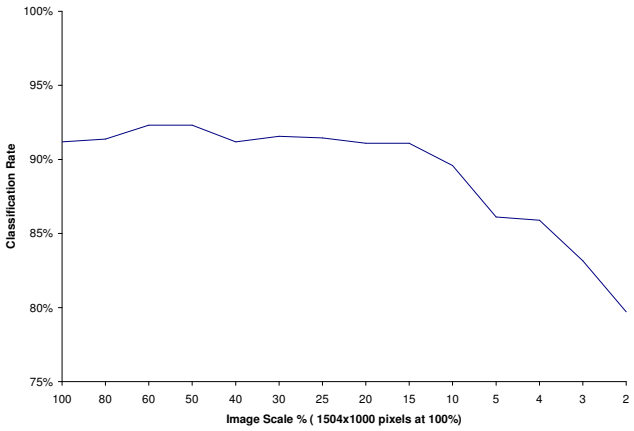

Figure 7.5: Degradation of classification as the size of the input pairs decreases. Classification remains level around 91% until the image is decreased to 10% of its original size (150 x 100 pixels). Decreasing the size of the images further to 2% of it original size (30 x 20 pixels) still results in a classification rate of 79.7%

(30 x 20 pixels) results in a classification rate of 79.7%.

This decrease of image size, in combination with the selection of features based on importance, begins to reduce the computation necessary for classification to the point where it becomes possible to do on-camera. On-camera classification would allow photographers to automatically organize panoramas, high-dynamic-range images, focal stacks, and super-resolution images as they are taken, significantly reducing the manual labor currently involved in their creation.

Investigating feature importance at smaller scales we see that many features are unaffected by the change in scale. The number of SIFT features, both matched and total per image, is a notable exception: as image size decreases the overall number of SIFT features and the number of matches drops significantly, even for super-resolution or panorama images, particularly when images are smaller than 10%. Additionally the focus overlap becomes much more important in distinguishing between classes, jumping in importance from #10 to #3.

7.3 Summary

In Chapter 7 we introduced two novel automatic registration systems that attempt to automatically classify registration problems based on the variation between image pairs. A rule based system was validated using a test set of 60 pre-classified image pairs verified by an independent user study of photographers. The system was able to identify 98% of the related ground truth pairs' main form of variation. 55% of pairs were correctly identified by a single form of variation allowing immediate selection of an algorithm. A further 32% of pairs proposed transforms were correctly selected using normalized cross correlation on the solution space of the proposed algorithms. Visual inspection of the final 13% of pairs suggests that the alignments proposed are superior to the 'correct' solution, however verification of this is difficult without ground truth alignments.

A second rule based system for classification of image pairs according to the category of registration they belong to was developed using support vector machines. 1100 pairs of images was collected, divided evenly amongst the five possible groupings: 220 Panorama pairs, 220 High-Dynamic-Range pairs, 220 Focal pairs, 220 Super-Resolution pairs, and finally 220 'unrelated' pairs, and is made available online to support future research [92].

A one to many classifier was trained which is able to classify between panoramas, high-dynamic-range-images, focal stacks, super-resolution images, and unrelated image pairs with a 91.18% accuracy. One to one classifiers were also developed to classify each of the categories individually. Classification rates for our one to one classifiers are as follows: Panorama image pairs are classified at 93.15%, high-dynamic-range pairs at 97.56%, focal stack pairs at 95.68%, super-resolution pairs at 99.25%, and finally unrelated image pairs at 95.79%.

The importance of features was investigated and the one to many classification rate was measured for feature vectors of various size, taken from a feature vector ordered by FScore. Classification was somewhat affected by the reduction in features, and use of the full feature vector is recommended for maximum accuracy.

Finally the invariance of the classification system towards the scale of the image used to calculate the feature vector was explored. Feature vectors were generated at: 100%, 80%, 60%, 50%, 40%, 30%, 25%, 20%, 15%, 10%, 5%, 4%, 3%, and 2% scale, and a one to many classifier was trained using 5-fold cross validation. The classification of our system remains level at ˜91% until the image is scaled to 10% of its original size (scaled to 150×100 pixels), suggesting that our feature vector is image size invariant within that

range. Decreasing the size of the images to 2% of their original size (30×20 pixels) results in a classification rate of 79.7%.

Improvement of the classifier through the development of a system which deals with sets of images, rather than image pairs, is left as future work. Image sets of a particular class are often taken in sequence, allowing the sequential use of our classifier to combine pairs into sets, however thought must be put into the system to prevent classification errors from compounding. This set based system would replace Brown's "Recognizing Panoramas" [61] providing a solution that is capable of ""Recognizing Panoramas, High-Dynamic-Range Images, Focal Stacks, and Super-Resolution images."

In addition, although our system focused on registration problems common to computational photography, extension into automatic registration for medical imaging and remote sensing would greatly benefit researchers. Such a system would require a greater degree of differentiation between problem types and would likely rely more heavily on image metadata to distinguish the variations between image pairs.

Finally, by combining our automatic detection of image registration problem type with the interpreter from our model of image registration we can accomplish the fully automatic solving of registration problems. Our automated system begins to approach the idea of the 'ultimate registration method' described by Zitová and Flusser at the end of their survey; a system able to recognize the type of task and to decide by itself about the most appropriate solution.

Feature	Panorama	HDR
Intensity(1)	105.415	91.5064
Intensity(2)	109.824	177.419
Hue(1)	23.3616	39.0234
Hue(2)	23.656	46.2149
Saturation(1)	154.964	115.74
Saturation(2)	153.859	74.4806
Power(1)	73.6296	38.6469
Power(2)	80.5668	37.3
Num Features(1)	26403	4982
Num Features(2)	27092	5395
Num Features Per Pixel(1)	0.0175552	0.0033125
Num Features Per Pixel(2)	0.0180133	0.0035871
Intensity Overlap	0.966408	0.40814
Hue Saturation Overlap	0.963122	0.41148
Focus Overlap	0.0596722	0.0180572
Matched Features	4467	594
Matched Features Per Pixel	0.002970	3.94947e-4
Matched-Feat Centroid (1)	(0.455,0.594)	(0.481,0.419)
Matched-Feat Centroid (2)	(0.487,0.548)	(0.423,0.427)
Intensity Overlap UL	0.0598358	0.0178404
Intensity Overlap UM	0.0597852	0.0178797
Intensity Overlap UR	0.0599069	0.0179156
Intensity Overlap ML	0.0598524	0.0179641
Intensity Overlap MM	0.0597819	0.0179501
Intensity Overlap MR	0.059732	0.0179681
Intensity Overlap LL	0.0598517	0.018018
Intensity Overlap LM	0.0596769	0.0180459
Intensity Overlap LR	0.059611	0.0180106

Table 7.5: Example of the feature vector and its corresponding values for a panorama image pair and a high-dynamic-range (HDR) image pair.

Type	Classification Rate
1:1 Panorama	93.15%
1:1 High-Dynamic-Range	97.56%
1:1 Focal Stack	95.68%
1:1 Super-Resolution	99.25%
1:1 Unrelated pairs	95.79%
1:Many Overall	91.18%

Table 7.6: Summary of classification rates for our one to many and one to one classifiers.

Ordered features	FScore
Hue / Saturation Overlap	2.524745
Intensity Overlap	2.022537
Matched Features	1.269260
Matched Features Per Pixel	1.269115
Intensity Overlap UR	1.199512
Intensity Overlap UM	1.197718
Intensity Overlap UL	1.196403
Intensity Overlap LL	1.195831
Intensity Overlap MR	1.193893
Focus Overlap	1.193237
Intensity Overlap MM	1.192491
Intensity Overlap ML	1.190727
Intensity Overlap LR	1.189334
Intensity Overlap LM	1.187563
Power(1)	0.364715
Power(2)	0.352534
Num Features Per Pixel(1)	0.266387
Num Features(1)	0.260805
Num Features Per Pixel(2)	0.234179
Num Features(2)	0.229223
Intensity(1)	0.227655
Saturation(2)	0.170803
Hue(2)	0.126434
Hue(1)	0.121199
Intensity(2)	0.069913
Saturation(1)	0.066153
Matched-Feat Centroid (2 X)	0.060326
Matched-Feat Centroid (1 X)	0.042694
Matched-Feat Centroid (2 Y)	0.006689
Matched-Feat Centroid (1 Y)	0.002026

Table 7.7: Ordered list of feature importance and corresponding FScore measure.

Type	# 1 Feature	# 2 Feature	# 3 Feature
1:1 Panorama	Power(1)	Power(2)	Matched Features
1:1 High-Dynamic-Range	Intensity Overlap	Intensity (1)	Intensity Overlap UR
1:1 Focal Stack	Power(1)	Power(2)	Matched Features
1:1 Super-Resolution	Matched Features	Matched Features Per Pixel	Intensity Overlap UR
1:1 Unrelated pairs	Hue / Saturation Overlap	Matched Features	Matched Features Per Pixel

Table 7.8: Top three features of each one to one classifier. As expected the most important features in each class relate directly to the common forms of variation that are indicative of that class.

Chapter 8

Towards a General Descriptive Model of Vision

"A different language is a different vision of life."

– Federico Fellini

In this chapter we extend the work on the creation of an interpretable model of image registration, presenting a path towards a general approach to problem-centric computer vision. This proposed model is called the Open Vision Language (OpenVL). Our desire with OpenVL is to provide access to sophisticated computer vision algorithms and problem spaces through descriptions of the problem. It is designed to allow a programmer to specify *what* it is they want to do, instead of *how* they want something done, providing a layer of abstraction above algorithmic details through the same means by which algorithmic details have been abstracted away from the problem of image registration. In addition to simplifying the developers implementation of computer vision applications, the abstraction also allows for code reuse, hardware acceleration, exploitation of advanced techniques and extensions to the language or algorithms.

Building upon the method developed for image registration, the OpenVL Framework consists of two components: models for describing vision problems, and a system for interpreting this description. We have explored these concepts for the problem of image registration in Chapters 4 and 6. Section 8.1 reexamines the conceptual elements of the language that allow more general vision problems to be described. These include the definition, representation, conditions, and finally expression of vision problems. We also explore how the conditions of vision problem, limitations on the solution, expectations and requirements, mechanisms for describing points and volumes in problem space, the properties of the image, and finally models, are used

to describe more complex problems. New concepts relating to the interpretation of the language are investigated in Section 8.2. The flexibility of the language allows for many possible methods of interpretation beyond those explored in Chapter 6, each with their own advantages and disadvantages.

8.1 The Open Vision Language

The Open Vision Language is a vernacular for describing computer vision problems. Unlike current vision *libraries* which provide only functionality in the form of specific algorithms, OpenVL provides a layer of abstraction that allows developers to use and combine these algorithms automatically based on the context of their problem.

In order to accomplish this automation, the description of the vision problem must be sufficiently detailed so that this context can be inferred. This section outlines in broad terms the concepts that must be covered in order to create a complete description of vision problems in general. In the preceding chapters we explored these concepts in detail as they applied to the task of image registration.

The fundamental units of description within the Open Vision Language are *models*, which are used to describe vision problems. Detailed in Chapters 3, 4, and 6 for the problem of image registration, the creation of an interpretable model for a given problem space is a significant undertaking. As we saw through our creation of an interpretable model of image registration however, these concepts combine to create the vocabulary through which vision problems can be more easily described by users inexperienced in vision. In this chapter we explore the generalization of this process, moving towards the creation of the OpenVL language.

Returning to Chapter 4 we see that the model of image registration is composed of four layers: Definition, Representation, Conditions, and finally Expression. A definition of our general model of computer vision is presented in Appendix B

8.1.1 Representation of Vision Problems

In order to know what type of problem is being solved, the desired solution must be specified. Even for a problem as specific as image registration, a number of possible transform types exist. Certain tradeoffs in dimensionality of the solution space allow less computationally intense algorithms to be selected: For example if the error in image registration is known to be based

Number of Faces	
Location.X	Location of the face in X, proportional to the image
Location.Y	Location of the face in Y, proportional to the image
Location.Z	Location of the face in Z, distance from camera in m

Table 8.1: Representation of the face detection problem space.

on x/y translation alone a much simpler and faster search can be performed along those two dimensions.

The representation of vision problems is problem dependant: the models or properties which make up the solution space of vision problems are determined by the problem itself. Some variation can exist within a field, such as whether face detection returns an (x,y) point in the image plane, or an estimation of the (x,y,z) coordinates, the sex of the subject, and whether they are smiling or not, however, and researchers must take care when developing a representation that they are not making implicit assumptions based on their usage or experience. This area has been explored in detail for image registration, and is left as future work for other researchers as individual problem areas are explored. For more complex problems such as object recognition the possible representations of the desired solution, both in the form of the object to be recognized and in the format of the solution, are even more diverse.

Specification of the limitations or restrictions that a developer has on the solution space is important as well. This information could potentially be useful in the selection of algorithms. It is also extremely important if constraint based algorithms algorithms are available. They can also help guide the selection of a solution in the instance where multiple solutions have been produced.

As an example of an alternate representation of a computer vision problem we present here our adaptation of a representation of face detection developed by Dr. Daesik Jang in conjunction with the author, Dr. Gregor Miller, and Dr. Sidney Fels. Our representation, presented in Table 8.1, allows developers to specify the type and range of solution they expect from the face detection algorithm and researchers to specify the range of the problem space which their algorithms are capable of solving.

8.1.2 Conditions of Vision Problems

The important pre existing conditions surrounding the problem must be specified by the developer in order to differentiate where their problem exists in n-D problem space. The design of these conditions is a critical process in the integration of any vision task into OpenVL. Understanding which conditions of the problem are important in the performance of a given algorithm is the responsibility of vision researchers who are experts in their field. By identifying these factors and allowing the developer to specify them they enable the use of sophisticated context specific algorithms to developers who would otherwise not have access because they lack expert knowledge of these methods. As an example, the conditions of a face detection problem might include the amount of occlusion expected in the images, allowing the system to determine whether or not occlusion invariant techniques will be required to detect faces.

As we saw in Section 4.2 when we explored the conditions related to image registration, these are often derived from and seem specific to the particular vision task being solved, however this can stem from an implicit assumption that these conditions do not exist in any other vision tasks. If the task in the above example were changed to object tracking then ideally the system will select an algorithm capable of tracking in spite of occlusion. While many of these problem spaces created by the intersection of different problem conditions are unlikely, examining vision in this manner does lead to some interesting new research opportunities when these assumptions aren't always true. More importantly it forces developers to think about any assumptions they might be making by identifying the conditions of the problem which affect the solution.

Table 8.2 presents the conditions drawn from the image registration model, *all of which* are applicable to vision problems in general. Recall that absolute conditions relate to a single image, while relative conditions can be thought of as relating between the image and the model used in the vision problem. Vision problems make the implicit assumption that none of these conditions vary between the image and model, however this is not always the case. Knowing if these conditions occur or not can lead to a better solution as different algorithms are likely to be more or less invariant across the different dimensions of the problem space that these conditions represent. The explicit specification of these conditions can also identify areas of future research where algorithms which work well under them need to be developed.

We present a prototype model for the conditions of the problem of face

Condition
Corrected Distortions
Translation
Rotation
Scale
Skew
Camera Extrinsics
Overlap
Uncorrected Distortions
Relative luminance
Focal Depth
Scene Lighting
Camera Intrinsics
Variations of Interest
Object Models
Object Motion

Table 8.2: Conditions of the image registration problem space that generalize to all vision problems.

registration in Table 8.3. Knowing these conditions allows for the selection of algorithms which are most appropriate, but does not change the solution space of face detection. For example if occlusion is known to be present a face detection algorithm which is occlusion invariant can be selected. Similarly if the developer is attempting to detect faces which are posed to face to the side, away from the camera, specific algorithms which are capable of detecting those poses can be selected.

Size of Face	Minimum and maximum size of face
Pose	Relative to camera-face pose
In-place Rotation	Rotation of the image
Occlusion of Face	Partial occlusion by other objects
Facial Conditions	Facial expression, age and gender
Imaging Conditions	Color, Illumination and so on

Table 8.3: Conditions of the face detection problem space.

8.1.3 Expressing Vision Problems

The solving of known vision problems by researchers with expert knowledge in the field is done on the basis of two main points. First, knowledge of the important input conditions of the problem is used to select an appropriate algorithm, since the conditions under which the problem is being solved significantly affects algorithm performance. Second, the representation of the desired solution helps determine exactly what type of problem is being solved. This aids not only in the selection of algorithms well suited to specific solution spaces, but also helps to set parameters within those spaces. In addition, any limitations on the solution space are similarly used to guide in the selection of an appropriate algorithm, and possibly also of an appropriate solution. Mimicking the process used by vision researchers, OpenVL relies on the same concepts in its interpretation of the developers problem description. This process of model expression and interpretation was initial developed for image registration in Chapters 4 and 6.

The representation and conditions surrounding a particular problem combine in a model which allows for the expression of the problem. In order to facilitate expression we have developed a number of key concepts, which were explored in Chapter 4. Expectations and requirements allow for the specification of expected and required representations and conditions. Properties allow for the expression of points, ranges, and volumes within the problem and solution space. Belief allows for the probabilistic weighting of properties. Finally models combine these concepts to allow for the specification of vision problems. These concepts hopefully will help researchers wishing to generalize their own problem domains as we strive towards a model of all of computer vision.

Models

Re-examining the concept of 'model,' we see that it varies significantly across problem domains. The application domain within this thesis, registration, does not require complex models, relying entirely on properties in order to represent the registration problem. The development of more complex models is explored briefly below and is a significant part of OpenVL, however a deep exploration of this concept is beyond the context of this thesis.

When a programmer wishes to specify that they require detection of "red" objects, a mechanism is required to let them describe what they mean by "red", at differing levels of complexity. At one end of the spectrum "red" could be the vector $(1, 0, 0)$ meaning a 100% red, 0% green and 0% blue mix-

ture of light, while at the other end it could be a complex gaussian mixture model specifying a multimodal probability density function of color components. Between these extremes could, for example, be distances around individual components in a Euclidean RGB space. Examples of other model specifications are shown in Table 8.4.

General language models that allow for the description of objects for the purpose of object detection or recognition may be difficult to fully achieve as ambiguities exist in the higher level language concepts that are desirable to use. While they can work for simple cases, the description of a complex scene is not yet feasible as we saw in our attempts to describe 'red' objects above. Once developed, however, such a language could become a sort of inverse to computer graphics, allowing developers to leverage tools and techniques in the computer graphics community to describe the objects or environments that are a part of their model.

An alternative approach would be to allow application developers to describe the dimension along which they are concerned, and to provide examples that meet their criteria within that context. In that scenario a developer trying to detect red balls would specify 'shape' and 'colour' as important dimensions, and then would provide examples of red balls under various conditions. This learning based approach, however requires the developer build a training set, which is a challenging task to do properly.

Problem specific models are more feasible to develop. Returning to our example of face detection, Table 8.5 below outlines the properties of the initial face detection model, and provides an example problem representation. Although the complexity of the problem space is higher than with image registration, the basic principles and techniques developed within this thesis have been utilized to create a similar model and problem centric mapping.

8.2 The OpenVL Language Interpreter

Chapter 6 presented a proof of concept interpreter capable of interpreting image registration models and selecting an appropriate algorithm. Due to the rich representation of expectations and requirements possible under our system, we anticipate a wide range of approaches for interpreting the context, ranging from simple case-based statements to highly sophisticated expert systems, probabilistic methods, and machine learning approaches. Of critical importance from the perspective of OpenVL as a language is that the choice of interpreter and vision algorithms are left to the discretion of the OpenVL vendor, allowing optimization, hardware acceleration, algorithm

Model	Describes	Example	Relationship		
			Abs	Rel	Dep
Appearance	Model of image region	Color, textures, templates	✓		
Color	PDF models of color regions	RGB space, Euclidean color regions, Gaussian mixture models, histograms	✓		
Depth	Specification of depth	Distance from viewpoint	✓	✓	
Difference	Metric space	Entropy, Euclidean, L_n norm		✓	
Focus	Focal properties	Degree of focus, point spread function	✓		✓
Geometry	Coordinate system, vertices	Euclidean and polar coordinates, rectified space, reference frames, clipping	✓		
Lighting	Lighting models	Specular, ambient, diffuse, emissive	✓		
Motion	Transformations over time	Shape translation over time		✓	
Regions	Enclosed areas of images	Shapes associated with image patches	✓		✓
Set	Set descriptions between regions	Intersection, union, subset of property models, i.e. focus, color, light, etc.			✓
Shape	Polygonal areas	Triangles, circles, polygons, ellipses	✓		✓
Transform	Mapping between spaces	Affine transforms, non-linear warps		✓	

Table 8.4: **OpenVL Models**: Illustrative examples (in alphabetical order) of model types for different context parameters used by blocks in the OpenVL state machine. Models can be defined: on a single image (Absolute), with respect to another (Relative), or dependent on relative properties between two images (Relative-Dependent). Models vary from from simple, such as RGB values for Color, to complex such as probability density functions (PDF) to support a large range of expression for expert and novices programmers.

Condition	Representation	Example
Size of Face	proportional to image width	0.075 - 0.5
Pose	Angles of roll, pitch and yaw	-15 - 15 degrees yaw
In-place Rotation	Angles of rotated image	-45 - 45 degrees
Occlusion of Face	Proportion of occlusion	0.2 (20%)
Facial Conditions	Labels for gender and age	F, 10-25
Imaging Conditions	Illumination and Color	10 lux

Table 8.5: The expression of face detection as a model.

evolution and tailored solutions by vision experts. One of the main advantages of OpenVL is that it does not prescribe any particular implementation to solve the requirements specified by application programmers. Thus, as hardware and software evolve, applications can immediately take advantage of them without the need to recode, or even recompile, much like computer graphics applications do not need to be recompiled when new graphics cards are developed.

We anticipate different groups will choose different interpretation methods. For example, one group may focus on rule based interpretation, using fast, hardware accelerated, efficient algorithms for registration, matching and decomposition that may not always perform well, but provide real-time analysis. For the other blocks in the machine which are less important to their target application, they may use software solutions. Likewise, a separate group concerned with very low errors for a wide range of registration problems may implement a variety of registration algorithms to provide an ensemble solution for that block using learning methods trained on a large data set of registration problems that allows them to accurately select one or more algorithms. Further, they may use the same approach for the other blocks as well.

Our intention within the context of this thesis is to provide an outline of the possible methods of interpretation, supplemented by an implementation of a simple proof of concept system seen in Chapter 6. By proving that it is possible to infer the problem from the developers description using OpenVL we provide a basis from which further research into the best method of interpretation can be explored.

8.2.1 Performance Evaluation Based (Direct) Interpretation

Problems for which measurable testbenches can be created allow for a direct mapping of their algorithms' performance. As we saw in Chapter 5 each of the image registration algorithms tested were directly evaluated under a variety of conditions, allowing for an inference of how well they should perform on similar problems should perform. The creation of a ground truth test set which covers a broad range of the entire image registration problem space was a significant undertaking and is not possible in all domains.

The expression of the image registration problem by a developer is either a point or a volume within the image registration problem space. In the case of a point a nearest neighbor or small volume about the point can be used, referring to the closest evaluated point in terms of problem conditions. Once the appropriate algorithms have been selected the system is capable of executing multiple algorithms and comparing their results. For volumes an averaging method is used, evaluating all points in the problem space that fit within that volume and averaging the algorithmic performance within the volume. Algorithms are then chosen based on the mean and standard deviation under the problem conditions. Selection based on the median of results within the volume may also be appropriate.

This method of interpretation allows for the best algorithm from similar situations to be used. Unfortunately as we saw the standard deviation of our alignment error is in some algorithms significant, bringing to question whether such a method of interpretation truly selects the best algorithm. Additionally, it is not always possible or feasible to develop testbenches for an entire problem space.

8.2.2 Rule Based and Learning Based Interpretation

The interpretation of which algorithm to use for a given problem can also be thought of as a classification problem, using the rich representation of expected and required representation and conditions possible under our system as a feature vector. The classification can either be one to one, with each algorithm independently determining whether it is able to solve the given problem, or multi-label classification, with one clear algorithm selected as the one that can best solve this problem. The classification can also be binary or can return a value. If one to one or non-binary classification are used then a second layer of logic is required to select either a single algorithm from amongst several candidates, or a single solution.

Deterministic methods, such as rule based expert systems, *probabilistic methods*, such as Bayesian systems, and *learning methods*, such as neural networks, support vector machines or case based reasoning systems, are all viable options for interpreting which vision algorithms to run. Learning methods are of particular interest since they could calculate algorithm performance estimates based on actual algorithm performance from our testbench, a technique likely to be more accurate than researchers own heuristic estimates.

Rule Based Methods

A rule based mapping of algorithmic performance within the image registration problem space could have been estimated, with each algorithm expressed as a volume or series of volumes within the image registration problem space using the model outlined Chapter 4. Where possible we envision researchers using a test set which covers a wide range of problems within their field, allowing for a direct mapping of their algorithms' performance, however as we saw in Chapter 5 the creation of a ground truth test set which covers an entire problem space is a significant undertaking and is not always possible for a given problem. This rule based method relies on the expert knowledge of researchers, who understand their algorithm well enough to be able to estimate where it will perform well, and where it is likely to fail to find a solution.

The expression of the image registration problem by a developer is either a point or a volume within the problem space. Algorithms whose volumes encompass or overlap these points or volumes are considered to be viable choices for finding a solution. Once the appropriate algorithms have been selected the system is capable of executing multiple algorithms and comparing their results.

Of particular importance in the estimation of these volumes is the mapping of the solution space, where multiple properties define the same solution space. If these representation properties are defined as ranges, then consideration of the combination of properties into all possible volumes within the solution space is necessary.

Probabilistic Methods

A series of inference rules can be established by specialists on a problem by problem basis that characterize the criteria of the input state representation that each candidate algorithm meets. Each algorithm returning an estimate

(from 0 to 1.0) of how well it will perform based on the description of the problem. A similar implementation based on joint probabilities that takes likelihood into account could be used to implement a bayesian selection system.

Both rule and Bayesian based methods require a vision expert to specify exactly how different state representations affect the quality of candidate algorithms, mapping out the performance of every algorithm in the n-dimensional problem space. In these situations we rely on the vision researcher integrating the algorithm to implement a function that takes the description of the problem and returns this estimate of performance. Unfortunately this type of self evaluation can lead to ties as comparative performance of algorithms is difficult to quantify without a testbench. This is particularly true when the estimation of different algorithms are not being done by the same vision experts. Other interpretation methods are unlikely to suffer from the same lack of discrimination in performance.

Learning Methods

The classification of algorithm based on learning methods is also possible. A neural network [35], support vector machine[17], or case based reasoning system [77] could be used to select which algorithm will perform best, either using example cases specified by experts to train the system to classify input state representations that fit each algorithm, or using the results from our testbench. This somewhat simplifies the requirements of specialists who wish to add new algorithms without understanding and modifying the existing system. Learning methods simply require a set of problem descriptions that accurately describes the conditions under which an algorithm performs well. This training set can be used to teach the system to make similar classifications when new descriptions are provided. Again this process relies on the vision researcher's ability to describe the conditions under which their algorithm performs well, and we suspect that the creation of a robust training set will require significant effort.

Finally, if normalized metrics are available to measure how well each algorithm has performed the vision task, then the learning based methods can be trained based on actual results. In this approach a series of test images, along with their corresponding state representation, are evaluated, either individually (one to one) or across all algorithms simultaneously (multi-label). This would allow OpenVL to make better algorithmic selections in cases where algorithms solve similar problems, and would be particularly useful in the evaluation of new algorithms. Assuming that a set of training data is

available that is representative of the problem domain, the algorithm can be automatically integrated into the system and used appropriately, possibly even utilizing unsupervised learning techniques to continue to improve its selection after initial training is complete. Unfortunately normalized metrics that are invariant to all possible conditions of the problem are rare.

8.2.3 Dealing with Multiple Candidate Algorithms

In our proof-of-concept example discussed in Chapter 6 we demonstrate the interpretation of registration. Our interpreter uses five registration algorithms that are combined and configured differently depending upon the context. For each of our registration algorithms we directly evaluate how well the algorithm will perform under the conditions described in the context, performing a value based one to one classification.

With this classification we can either select whichever is indicated as the best choice, run all candidate algorithms that exceed a certain threshold such as mean plus variance and pick the best result, selecting from amongst the algorithms, or run all candidate algorithms that exceed a certain threshold and select from amongst their solutions. It is also possible to combine the results, however only makes sense in specific problem domains. Selecting from amongst several solutions requires a normalized metric to measure how well the algorithm has performed the vision task.

In our proof of concept system, if an algorithm is deemed appropriate for the current data then it will be executed with the values and data specified by the application through the context. Each registration algorithm returns a solution within the requirements specified by the application, unless none fit in which case it returns failure.

As discussed in Chapter 6, because these algorithms may have different error spaces, each algorithm's performance must be normalized. This can be done by calculating the registration error of each transform on all candidate algorithms.

8.2.4 Adding New Algorithms to the Interpreter

The addition of new algorithms to the OpenVL interpreter can be a significant process, depending on the type of interpreter. Creators of interpreters should be aware of the issues that algorithm developers will face when adding new algorithms. Below we explore the method of adding new algorithms for each of the interpreters described.

Deterministic Methods

New algorithms can be easily integrated into deterministic methods, assuming a testbench for the field has been established across the problem space. By evaluating the new algorithm directly at each point its performance can be determined, and it can be selected immediately by a deterministic interpreter if it outperforms the existing algorithms. This is one of the significant advantages of deterministic methods.

Rule Based and Probabilistic Methods

Adding new algorithms to either of these systems can be a complex process, requiring a vision expert to estimate the performance of the new algorithm across the entire n-dimensional problem space. This is fortunately not as complex as it initially seems as the majority of algorithms, at least in the case of image registration, are designed to work under a specific set of input conditions, and are often assumed to perform poorly if those conditions are not met. This assumption is not always true and a more robust method directly tied to the performance of each algorithm across the entire problem space is desirable. Deterministic and probabilistic methods do however have the advantage that the knowledge embedded within the system is clearly defined and is more understandable.

Learning Methods

The addition of algorithms to learning based methods requires a series of models which cover the dimensions of the n-dimensional problem space that the algorithm is capable of solving. A one to one classifier can then be trained to detect when these or similar conditions arise. A one to many classifier can also be trained by combining the data from all algorithms, although its accuracy at classifying is likely to be limited if algorithms do not have some sort of accurate performance estimate beyond whether it 'works' or not. As with rule based and probabilistic methods this method relies on the algorithm developer to accurately describe the conditions and representations where their algorithm performs well, which they may not necessarily know beyond.

8.3 Summary

In this chapter we have extented the model of image registration introduced in Chapter 4, providing a framework for developing models of other com-

puter vision problems. A second computer vision problem space of face detection was presented and a basic representation, conditions, and expression were shown to demonstrate how the process used in the creation of a model for image registration can be used in other computer vision fields. In addition, a variety of interpreters were proposed, and the implications of using these interpretation types was explored, guiding researchers who wish to create their own OpenVL interpreter, whether for image registration or for a problem domain of their own. It is our hope that researchers in other fields will follow the approach presented in this thesis and create models of their own problem domains, growing the body of computer vision domains that OpenVL covers.

Chapter 9

Conclusion and Future Work

"The obscure we see eventually, the completely apparent takes longer."

– Edward R. Murrow

Computer Vision is the study of how computers and machines see and understand the world. This 'understanding' is achieved by creating models or representations of a scene, however as we saw in Chapter 2 under the current framework sophisticated expert knowledge is required to understand and properly utilize the internal models used in order to effectively make use of these algorithms. Researchers must understand the vision task and the conditions surrounding their problem, and only through significant research efforts select an appropriate algorithm which will solve the problem most effectively under these constraints.

Within this thesis we have presented a new taxonomy for image registration based on this same understanding of the vision task and the conditions surrounding the problem, with our model of the problem space providing an abstraction layer over image registration algorithms. This style of problem centric computer vision allows programmers who are not vision researchers to access advanced image processing techniques without requiring specific knowledge of the underlying algorithms that implement them. It also allows improved algorithms to seamlessly replace older implementations, including graphics or potentially vision card based implementations, providing programmers using a problem centric software library with an instantaneous upgrade path without reprogramming or integrating a new implementation.

Vision researchers who develop algorithms also see significant benefits to the widespread understanding of this knowledge. First, they can identify and represent the problem conditions under which their algorithms are being evaluated, allowing for much more robust comparison of algorithms.

The existence of a problem space which models all of the possible conditions under which a vision task may be performed allows for the creation of test sets which span well defined problem conditions providing a more direct comparison of performance. Second, researchers can evaluate the conditions under which their algorithms perform well, providing a deeper understanding of performance and potentially providing insights into how it may be improved. Third, by examining the problem space and the existing algorithms that support it researchers can identify niches within the problem space which are useful but do not have solutions. Finally, the description of the vision problem itself is represented in such a way that non-vision experts can understand making the algorithms much more accessible and usable outside of the vision research community.

Extending beyond image registration, our proposed general framework of vision is designed to make all vision tasks more accessible to developers by providing a model of vision which allows for the description of *what* the developer wants to achieve without requiring the specification of *how* the problem is solved. In order to provide this accessibility a common model for the significant conditions surrounding a given vision problem must be established, and the representation of the solution space must be well defined. This is a difficult task and requires an in depth understanding of the field, however once established,

Reorganizing computer vision in this way requires a deep understanding of individual vision problems. In this thesis we have focused on image registration problems, providing a starting point for our proposed descriptive language model of vision, OpenVL. Image registration was chosen because it is a mature problem with a wide range of solutions that work well under specific conditions. This exploration provides a pathway for further development of our language through expansion into other areas of vision.

9.1 Contributions

To summarize, the contributions of this thesis are as follows:

- First, an up-to-date survey of image registration techniques was carried out using an existing taxonomy of image registration.
- From this initial mapping a new taxonomy was developed and existing techniques were again mapped according to the new taxa.
- A model of the image registration problem space was developed based

on this taxonomy, allowing vision researchers and non experts to describe image registration problems in a well defined manner.

- A testbench was developed for the image registration problem space, and several image registration methods were evaluated. The impact of each testbench parameter on the alignment error and success ratio of the algorithms were also determined.
- A proof of concept method of interpretation was developed which allows for the selection of appropriate algorithm(s) based on a given problem space description.
- Two methods of automatic detection of common types of image registration problems were developed, allowing for the automatic classification of image registration problems.
- Finally, the methodology used to create model for image registration was extrapolated providing a starting point for a general model of computer vision.

We review each of those in detail below, before touching on future work. Publications related to this thesis include: [40, 64, 65, 72, 73, 90]

An up to date Survey of Image Registration Techniques

Chapter 2 provides a literature review of both image registration techniques and of computer vision libraries and frameworks. Image registration techniques are presented under the traditional algorithm centric taxonomy, and the limitations of current methods of evaluation are discussed, motivating the development of our novel taxonomy, model, and testbench. Existing and past vision libraries and frameworks are also examined in detail and contrasted with the proposed problem-centric methodology.

A New Taxonomy of Image Registration

In Chapter 3 we developed a mapping of the image registration problem domain which focuses on the types of variation that occur between images to be registered. These forms of variation are presented as dimensions of the image registration problem, providing an abstraction which allows us to think of algorithms as supporting volumes within the n-dimensional problem space. Existing algorithms have been introduced into this mapping according to the different forms of variation that they have been designed

to support, including instances where algorithms support forms of variation beyond those they are traditionally used for. The reorganization of image registration into our problem-centric taxonomy provided a basis for the development of a model of image registration.

A Model of Image Registration

Using the new variation centric taxonomy of image registration, Chapter 4 created a model allowing for the specification of image registration problems. In Appendix A we provided a formal *definition* of image registration and extend this definition into the applied domain. In Section 4.1 we explored the *representation* of the inputs and outputs used in the problem of image registration, allowing developers and researchers to specify the type of solution that they are expecting. Section 4.2 explored the different *conditions* of the problem of image representation, presented as forms of variation. Section 4.3 introduced the necessary concepts and types used in our model, and provided a mapping of the representations and conditions of registration into a formal model through which image registration can be expressed. Finally Section 4.4 demonstrated several common image registration problems under our model. These layers represent our framework of accessible computer vision.

A Testbench for Image Registration

In Chapter 5 a method for creating image registration testbenches was detailed, which uses synthetic variations and a ground truth transform to create image registration pairs with known transform and image parameters. Three testbenches of 5K image pairs were created, each of which covers a range of problems with a common form of variation. No variation, exposure variation, and focus variation regions of the image registration problem space were tested in addition to variations in image size, exposure, and transform parameters.

Our testbench tests four algorithms that cover a variety of registration methods: a gradient descent intensity-based method [70]; a modification of a median-based method [100] that performs gradient descent on binary maps of the images' median values; a mutual-information-based method [60]; and finally a SIFT feature-based method [61] which uses RANSAC [28] to solve for alignment. Although we eventually plan to add more, these methods are sufficient to validate the testbench.

Analysis of variance was used to measure the effect of each of the test pa-

rameters on the alignment error of the algorithms tested, while a χ^2 method was used to measure the effect on success ratio. From this initial investigation several significant parameters were identified, and their impact within these areas of the problem space was directly explored. The results of the testbench were somewhat surprising. The range of the image registration problem space the our SIFT feature based implementation was able to cover was much greater than expected. Although its performance breaks down in a number of regions it is much more capable of dealing with variation than expected. The other algorithms tested did not perform as well as expected, possibly due to difficulties of the gradient descent algorithm at solving the six dimensional affine transform. Still, for a number of areas within the image registration problem space the Lucas Kanade based method proved superior. This testbench provides a basis from which the limitations due to transform parameters, image size, overlap, exposure variation, and focus variation of other algorithms can be explored.

Automatic Classification

In Chapter 7 we introduced two novel automatic registration systems that attempt to automatically classify registration problems based on the variation between image pairs. A rule based system was validated using a test set of 60 pre-classified image pairs verified by an independent user study of photographers. The system was able to identify 98% of the related ground truth pairs' main form of variation. 55% of pairs were correctly identified by a single form of variation allowing immediate selection of an algorithm. A further 32% of pairs proposed transforms were correctly selected using normalized cross correlation on the solution space of the proposed algorithms. Visual inspection of the final 13% of pairs suggests that the alignments proposed are superior to the 'correct' solution, however verification of this is difficult without ground truth alignments.

A second learning-based system for classification of image pairs according to the category of registration they belong to was developed using support vector machines. 1100 pairs of images was collected, divided evenly amongst the five possible groupings: panorama, high-dynamic-range, focal, super-resolution, and finally 'unrelated.'

A one to many classifier was trained which is able to classify between category with a 91.18% accuracy. One to one classifiers were also developed to classify each of the categories individually. Classification rates for our one to one classifiers are as follows: Panorama image pairs are classified at 93.15%, high-dynamic-range pairs at 97.56%, focal stack pairs at

95.68%, super-resolution pairs at 99.25%, and finally unrelated image pairs at 95.79%.

The importance of features was investigated and the one to many classification rate was measured for feature vectors of various size, taken from a feature vector ordered by FScore. Classification was somewhat affected by the reduction in features, and use of at least seven features in the feature vector is recommended for maximum accuracy.

Finally the invariance of the classification system towards the scale of the image used to calculate the feature vector was explored. Feature vectors were generated across different scales, one to many classification was performed at each level. The classification remains level at ˜91% until the image is scaled to 10% of its original size (scaled to 150 × 100 pixels), suggesting that our feature vector is image size invariant within that range. Decreasing the size of the images to 2% of their original size (30 x 20 pixels) results in a classification rate of 79.7%.

Mapping our automatic detection of image registration problem type to a volume within the image registration problem space which corresponds with the example problem pairs used to create each category, we can use our interpreter, introduced in Chapter 6 to automatically solve registration problems using the appropriate algorithm. This level of automation begins to approach the 'ultimate registration method' described by Zitová and Flusser; a system able to recognize the type of registration task and to decide by itself about the most appropriate solution.

A General Model of Computer Vision

Finally, in Chapter 8 we generalized the model of image registration introduced in Chapter 4 into a general model of computer vision. A second computer vision problem space of face detection was presented and a basic representation, conditions, and expression were shown to demonstrate how the process used in the creation of a model for image registration can be used in other computer vision fields. In addition, a variety of interpreters were proposed, and the implications of using these interpretation types was explored, guiding researchers who wish to create their own OpenVL interpreter, whether for image registration or for a problem domain of their own. It is our hope that researchers in other fields will follow the approach presented in this thesis and create models of their own problem domains, growing the body of computer vision domains that OpenVL covers.

9.2 Future Work

Image registration remains one of the most important tasks in computer vision when combining information from various sources. Chapter 2 gave an up to date survey of image registration techniques, building from previous surveys [14, 80, 93, 103] using the existing framework of the field. The reorganization of image registration into our variation-centric taxonomy, and model derived from that reorganization highlighted several additional research opportunities that could significantly advance the field.

Exploring the concept of the different types of variation common to image registration, we found that rather than isolated problem spaces, these variations represent different dimensions of a single problem space. Most image registration methods are designed to work along a single main dimension, however the combination of these is becoming more common, particularly in sensor / structure combinations for multimodal non-rigid medical imaging. Another notable multidimensional example is Schechner and Nayer's HDR panorama stitching method [87]. The examination of other combinations of variation such as focus and structure, or exposure and sensor could prove interesting, although many of these higher dimensional pairings are likely best solved by examining the groups of images as a whole and choosing which pairs to match using conventional methods, a topic not covered within this thesis. Taking the concept of a standard model of image registration further, we envision the creation of image registration methods which take our model of image registration as input and adjust their parameters or even their algorithm accordingly, actively using the knowledge of the problem to aid in its solution.

We also saw how each registration algorithm can be represented as an n-dimensional volume within this multidimensional space, outlining its utility under various conditions. Such a representation, either specified by the algorithms' creator or derived through a testbench, gives researchers who require registration as a part of their system a better sense of which algorithms are likely to work under the conditions of their problem. The testbench established in Chapter 5 provides a starting point for the understanding the impact that these conditions have on the performance of different registration algorithms. In order to enable a deeper level of understanding a number of steps must be undertaken. First, synthetic variations for each of the other dimensions of the problem space not covered by the current testbench must be developed in order to allow for the creation of image pairs with all forms of variation seen under our model. The investigation into other forms of variation not yet covered by the model may also be

necessary. In order to extend the methodology to include image registration within the field of medical imaging significant research must be undertaken to establish a model which encapsulates the new conditions that occur in that problem domain. Another important area of future research will be in the investigation of interactions between the various forms of synthetic variation. With the ability to create image pairs which span the entire volume of the image registration problem space it begins to become possible to create a testbench which accurately measures the performance of image registration algorithms. Using this improved testbench other image registration algorithms must be evaluated and explored in order to establish a suite of algorithms useful in different situations. The sensitivities of each algorithm to the various parameters of our testbench reveal a lot about their limitations, and could lead to the development of better algorithms. In addition to other algorithms, the effect of algorithmic parameters on performance under different conditions must be explored. The combination of the parameter space of an algorithm and of the image registration problem space is a high dimensional space, requiring significant computing resources, however a robust evaluation of performance need only be done once, making it possible to investigate. This investigation would have to be undertaken in close conjunction with the algorithm developer however, as the internal parameters are often abstract and the appropriate range, distribution, and combination of parameter settings is generally better known by them. Finally, research into the density of the testbench at various points across the problem space should be investigated. Volumes which coincide with common image registration problems should be investigated in detail, while regions at the extremes of variation interaction can be examined in less detail.

From this improved testbench more sophisticated interpretation techniques, such as those proposed in Chapter 8 can be developed. As more is known about the image registration problem space, and algorithms relative performance within it, it becomes possible to create more accurate interpreters. A direct evaluative interpreter was utilized within this thesis, however a number of other interpreter methods may improve the overall selection process, or may be more appropriate for different problem types. The use of rule based, learning based, or probabilistic methods is therefore highlighted as an area for future research, particularly where testbenches to quantitatively measure performance are not available.

A more robust testbench and interpreter would motivate the creation of new automatic classification techniques which are designed to identify regions of the image registration problem space which require further differentiation in order to select the most appropriate algorithm. The combination

of this automatic method of classification, a robust interpreter, and a test-bench which covers the entire image registration problem space would finally allow for the creation of the 'ultimate registration method' proposed by Zitová and Flusser at the end of their survey [104]. The system would be able to recognize the type of image registration task and decide by itself about the most appropriate solution, relieving developers of the burden of classifying their image registration problems. An expert system based approach and a support vector machine based approach have been explored for the purposes of classification, however a number of other learning or Bayesian based methods may improve upon their performance. The development of new features which measure differences between the image pairs may also provide a means of improvement of the classification methods.

Finally, the methods developed here to create a model and interpreter for image registration must be expanded into other problem areas. The creation of a model for face detection introduced in Chapter 8 is a start, but our goal is the creation of models which span the majority of problems in computer vision. The development of such a system across the computer vision problem space will require significant innovation in the form of detailed problem models, new and more sophisticated methods of interpretation, and new methods of classification. This is the work of the entire community of vision researchers, and could lead to advancements in the field similar to those seen in computer graphics following the creation of OpenGL. It is hoped that this thesis provides a starting point for such a significant undertaking, highlighting our philosophy, our initial approach to the problem, and the challenges that researchers who choose to follow may face.

Bibliography

[1] Amir Afrah, Gregor Miller, Donovan Parks, Matthias Finke, and Sidney Fels. Hive: A distributed system for vision processing. In *Proceedings of the International Conference on Distributed Smart Cameras*, September 2008.

[2] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. In *Proceedings of SIGGRAPH*, pages 294–302, New York, NY, USA, 2004. ACM Press.

[3] Apple Developer - Developing with Core Image. http://developer.apple.com/macosx/coreimage.html.

[4] Apple Quartz Composer. http://developer.apple.com/ graphicsimaging/quartz/quartzcomposer.html.

[5] Pietro Azzari, Luigi Stefano, and Stefano Mattoccia. An evaluation methodology for image mosaicing algorithms. In *Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 89–100, Berlin, Heidelberg, 2008. Springer-Verlag.

[6] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[7] A. Bardera, M. Feixas, and I. Boada. Normalized similarity measures for medical image registration a. bardera, m. feixas and i. boada. In *SPIE Journal of Medical Imaging*, pages 108–118, 2004.

[8] 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, Apr 2002.

[9] F.L. Bookstein. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, Jun 1989.

[10] S. Borman and R. Stevenson.

[11] Andrew P. Bradley, Michael Wildermoth, and Paul Mills. Virtual microscopy with extended depth of field. In *Proceedings of the Digital Image Computing on Techniques and Applications*, page 35, Washington, DC, USA, 2005. IEEE Computer Society.

[12] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., 1st edition, October 2008.

[13] Morten Bro-Nielsen and Claus Gramkow. Fast fluid registration of medical images. In *VBC '96: Proceedings of the 4th International Conference on Visualization in Biomedical Computing*, pages 267–276, London, UK, 1996. Springer-Verlag.

[14] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24:325–376, 1992.

[15] Camellia. http://camellia.sourceforge.net/.

[16] James B. Campbell. *Introduction to remote sensing*. Guildford Press, 4th edition, 2008.

[17] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[18] Y. W. Chen and C. J. Lin. *Combining SVMs with various feature selection strategies*. Springer, 2006.

[19] G.E. Christensen, R.D. Rabbitt, and M.I. Miller. Deformable templates using large deformation kinematics. *IEEE Transactions on Image Processing*, 5(10):1435–1447, Oct 1996.

[20] Régis Clouard, Abderrahim Elmoataz, Christine Porquet, and Marinette Revenu. Borg: A knowledge-based system for automatic generation of image processing programs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:128–144, February 1999.

[21] W R Crum, T Hartkens, and D L G Hill. Non-rigid image registration: theory and practice. *British Journal of Radiology*, 77(2):140–153, 2004.

[22] E. D'Agostino, Frederik Maes, Dirk Vandermeulen, and Paul Suetens. A viscous fluid model for multimodal non-rigid image registration using mutual information. In *Proceedings of the 5th International Conference on Medical Image Computing and Computer-Assisted Intervention-Part II*, pages 541–548, London, UK, 2002. Springer-Verlag.

[23] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH*, New York, NY, USA, 1997. ACM.

[24] L. Ding, A. Goshtasby, and M. Satter. Volume image registration by template matching. *Image and Vision Computing*, 19(12):821 – 832, 2001.

[25] A. L. Drozd, A. C. Blackburn, I. P. Kasperovich, P. K. Varshney, M. Xu, and B. Kumar. A preprocessing and automated algorithm selection system for image registration. volume 6242, page 62420T. SPIE, 2006.

[26] J. Ens and P. Lawrence. An investigation of methods for determining depth from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:97–108, 1993.

[27] O. Firschein and T. M. Strat. *Radius: Image Understanding For Imagery Intelligence*. Morgan Kaufmann, 1997.

[28] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of ACM*, 24(6):381–395, June 1981.

[29] Jan Flusser, Barbara Zitová, and Toms Suk. Invariant-based registration of rotated and blurred images. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium.*, pages 1262–1264. IEEE Computer Society, 1999.

[30] 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, Sep 1991.

[31] Rui Gan, Jue Wu, Albert C. S. Chung, Simon C. H. Yu, and William M. Wells Iii. Multiresolution image registration based on kullback-leibler distance. In *In The 7th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI04*, pages 599–606. SpringerVerlag, 2004.

[32] Gandalf. http://gandalf-library.sourceforge.net/.

[33] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, pages 147–151, 1988.

[34] Stefan Henn and Kristian Witsch. Multimodal image registration using a variational approach. *SIAM Journal of Scientific Computation*, 25(4):1429–1447, 2003.

[35] John Hertz, Richard G. Palmer, and Anders S. Krogh. *Introduction to the Theory of Neural Computation*. Perseus Publishing, 1991.

[36] Image Magick. http://www.imagemagick.org.

[37] M. Irani and P. Anandan. Robust multi-sensor image alignment. pages 959–966, Jan 1998.

[38] GR Iversen and H Norpoth. *Analysis of variance*. Sage Publicalions. Inc., 1987.

[39] Ralph E. Jacobson, Sidney F. Ray, Geoffrey G. Atteridge, and Norman R. Axford. *The Manual of Photography: Photographic and Digital Imaging 9th Ed.* Oxford: Focal Press, 2000.

[40] Daeshik Jang, Gregor Miller, Sidney Fels, and Steve Oldridge. A user oriented language model for face detection. *IEEE Workshop on Person Oriented Vision*, January 2011.

[41] K.P. Maher J.C.P. Heggie, N. A. Liddell. Applied imaging technology. *Australasian Physical and Engineering Science in Medicine*, 25:87–87, 2002.

[42] Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High dynamic range video. In *Proceedings of SIGGRAPH*, pages 319–325, New York, NY, USA, 2003. ACM.

[43] Yan Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. volume 2, pages II–506–II–513 Vol.2, June-2 July 2004.

[44] Kerr, Doug. APEX - The Additive System of Photographic Exposure.

[45] J J Koenderink and A J van Doom. Representation of local geometry in the visual system. *Journal of Biology and Cybernetics*, 55(6):367–375, 1987.

[46] Charles Kohl and Joe Mundy. The development of the image understanding environment. In *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 443–447. IEEE Computer Society Press, 1994.

[47] Konstantinos Konstantinides and John R. Rasure. The khoros software development environment for image and signal processing. *IEEE Transactions on Image Processing*, 3:243–252, 1994.

[48] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using affine-invariant regions. volume 2, pages II–319–II–324 vol.2, June 2003.

[49] Thomas M. Lillesand and Ralph W. Kiefer. *Remote Sensing and Image Interpretation*. Wiley, 6th edition, 2007.

[50] J. Liu, B.C. Vemuri, and F. Bova. Multimodal image registration using local frequency. pages 120–125, 2000.

[51] J. Liu, B.C. Vemuri, and J.L. Marroquin. Local frequency representations for robust multimodal image registration. *IEEE Transactions on Medical Imaging*, 21(5):462–469, May 2002.

[52] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[53] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision (darpa). In *Proceedings of the 1981 DARPA Image Understanding Workshop*, pages 121–130, April 1981.

[54] 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, April 1997.

[55] J. Maintz and M. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.

[56] Alexei Makarenko, Alex Brooks, , and Tobias Kaupp. On the benefits of making robotic software frameworks thin. In *International Conference on Intelligent Robots and Systems*, 2007.

[57] T. Makela, P. Clarysse, O. Sipila, N. Pauna, Quoc Cuong Pham, T. Katila, and I.E. Magnin. A review of cardiac image registration methods. *IEEE Transactions on Medical Imaging*, 21(9):1011–1021, Sept. 2002.

[58] Songrit Maneewongvatana and David M. Mount. The analysis of a probabilistic approach to. In *In Proceedings of the 2001 Workshop on Algorithms and Data Structures*, pages 276–286, 2001.

[59] Takashi Matsuyama and Vincent Hwang. Sigma: a framework for image understanding integration of bottom-up and top-down analyses. In *Proceedings of the 9th international joint conference on Artificial intelligence - Volume 2*, pages 908–915, San Francisco, CA, USA, 1985. Morgan Kaufmann Publishers Inc.

[60] David Mattes, David R. Haynor, Hubert Vesselle, Thomas K. Lewellen, and William Eubank. Nonrigid multimodality image registration. *Proceedings of SPIE.*, 4322:1609–1620, 2001.

[61] M.Brown and D. G. Lowe. Recognising panoramas. *Proceedings of the Ninth IEEE International Conference on Computer Vision*, 2:1218–1225, 16-16 Oct. 2003.

[62] Tim McInerney and Demetri Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91 – 108, 1996.

[63] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, October 2005.

[64] Gregor Miller, Sidney Fels, and Steve Oldridge. Axioms of computer vision. *Canadian Conference on Computer and Robot Vision*, May 2011.

[65] Gregor Miller, Steve Oldridge, Daeshik Jang, and Sidney Fels. An automated problem-to-algorithm mapping using a description model. *IEEE Workshop on Workshop on Person Oriented Vision*, January 2011.

[66] Mehran Moshfeghi. Elastic matching of multimodality medical images. *CVGIP: Graph. Models Image Process.*, 53(3):271–282, 1991.

[67] Joseph Mundy. The image understanding environment program. *IEEE Expert: Intelligent Systems and Their Applications*, 10(6):64–73, 1995.

[68] National Alliance for Medical Image Computing. http://www.itk.org/.

[69] National Alliance for Medical Image Computing. http://www.na-mic.org/Wiki/index.php/NA-MIC-Kit.

[70] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, August 1999.

[71] Michael J. Ocean, Azer Bestavros, and Assaf J. Kfoury. snbench: programming and virtualization framework for distributed multitasking sensor networks. In *Proceedings of the 2nd international conference on Virtual execution environments*, pages 89–99, New York, NY, USA, 2006. ACM.

[72] Steve Oldridge, Gregor Miller, and Sidney Fels. Automatic classification of image registration techniques. In *Proceedings of the International Conference on Computer Vision*, October 2009.

[73] Steve Oldridge, Gregor Miller, and Sidney Fels. Classification of image registration problems using support vector machines. *IEEE Workshop on Applications of Computer Vision*, January 2011.

[74] Steve Oldridge, Gregor Miller, and Sidney Fels. A model for image registration. *Canadian Conference on Computer and Robot Vision*, May 2011.

[75] Steve Oldridge, Gregor Miller, and Sidney Fels. A testbench for image registration. *Submitted to International Conference on Computational Photography*, April 2011.

[76] Parker, Fred. http://www.fredparker.com/.

[77] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[78] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance,

and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[79] J Peng, KL Lee, and GM. Ingersoll. An Introduction to Logistic Regression Analysis and Reporting, 2007.

[80] J.P.W. Pluim, J.B.A. Maintz, and M.A. Viergever. Mutual-information-based registration of medical images: a survey. *Medical Imaging, IEEE Transactions on*, 22(8):986–1004, Aug. 2003.

[81] Arthur R. Pope and David G. Lowe. Vista: A software environment for computer vision research. *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[82] E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec. *High Dynamic Range Imaging. Data Acquisition, Manipulation, and Display.* Morgan Kaufmann, 2005.

[83] G.K. Rohde, A. Aldroubi, and B.M. Dawant. The adaptive bases algorithm for intensity-based nonrigid image registration. *IEEE Transactions on Medical Imaging*, 22(11):1470–1479, Nov. 2003.

[84] D. Rueckert, L.I. Sonoda, C. Hayes, D.L.G. Hill, M.O. Leach, and D.J. Hawkes. Nonrigid registration using free-form deformations: application to breast mr images. *IEEE Transactions on Medical Imaging*, 18(8):712–721, Aug. 1999.

[85] Peter Sand and Seth Teller. Video matching. *ACM Transactions of Graphics*, 23(3):592–599, 2004.

[86] Frederik Schaffalitzky and Andrew Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 414–431, London, UK, 2002. Springer-Verlag.

[87] Yoav Y. Schechner and Shree K. Nayar. Generalized mosaicing: High dynamic range in a wide field of view. *International Journal of Computer Vision*, 53(3):245–267, 2003.

[88] ShapeLogic. http://www.shapelogic.org.

[89] Ravi K. Sharma and Misha Pavel. Multisensor image registration. *Journal of the Society for Information Display*, pages 951–954, 1997.

[90] Changsong Shen, Steve Oldridge, and Sidney Fels. Open source vision library (openvl) based local positioning system. *IEEE Conference on Advanced Video and Signal Based Surveillance*, 0:105, 2006.

[91] Noah Snavely, Rahul Garg, Steven M. Seitz, and Richard Szeliski. Finding paths through the world's photos. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*, 27(3):11–21, 2008.

[92] Sidney Fels Steve Oldridge, Gregor Miller. http://www.ece.ubc.ca/ hct/registration/.

[93] Richard Szeliski. Image alignment and stitching: a tutorial. *Foundional Trends in Computational Graphics and Visualization*, 2(1):1–104, 2006.

[94] Anna Tomaszewska and Radoslaw Mantiuk. Image registration for multi-exposure high dynamic range image acquisition. In *Proceedings of the International Conference of Central Europe on Computer Graphics, Visualization, and Computer Vision*, 2007.

[95] Kari Torkkola. Feature extraction by non parametric mutual information maximization. *Journal of Machine Learning Research*, 3:1415–1438, 2003.

[96] Godfried T. Toussaint. A simple linear algorithm for intersecting convex polygons. *The Visual Computer*, 1:118–123, 1985.

[97] L. Van Gool, T. Moons, and M. Proesmans. Mirror and point symmetry under perspective skewing. pages 285–292, Jun 1996.

[98] Paul Viola and Wells. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[99] VXL. http://vxl.sourceforge.net/.

[100] Greg Ward. Robust image registration for compositing high dynamic range photographs from handheld exposures. *Journal of Graphics Tools*, 8:17–30, 2003.

[101] Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. SATzilla: Portfolio-based Algorithm Selection for SAT. *Journal of Artificial Intelligence Research*, 32(1):565–606, 2008.

[102] Gehua Yang, C.V. Stewart, M. Sofka, and Chia-Ling Tsai. Registration of challenging image pairs: Initialization, estimation, and decision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1973–1989, Nov. 2007.

[103] Barbara Zitová and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21:977–1000, 2003.

[104] Barbara Zitová, Jaroslav Kautsky, Gabriele Peters, and Jan Flusser. Robust detection of significant points in multiframe images. *Journal of Pattern Recognition*, 20(2):199–206, 1999.

Appendix A

A Formal Definition of Image Registration

In order to create a formal definition of image registration, a number of other concepts must be defined. These concepts begin to outline a more formal representation for the essential aspects of computer vision such as scenes, views, cameras, images, features, and their relationship to one another. The extension of this set theory based representation into a model for all of computer vision is introduced in Chapter 8. Once we have a definition of key vision components we can then formally define the problem of image registration.

A.1 Notation and Basic Definitions

We will use set notation to define our computer vision problems and components. A superscript denotes the view number which that item refers to, such as V^n (in this case, the nth view). A subscript denotes the item number, such as o_k (the kth object).

We use the notation $\mathbb{N}_N$ to represent the set $\{1, \ldots, N\}$ and $\mathbb{W}_N$ to represent the set $\{0, \ldots, N\}$. For intervals, we use square brackets - $[,]$ - for inclusive and round brackets for - $(,)$ - exclusive. Both may be used to define one interval e.g. $[0, 1)$ is the interval from 0 to 1 including 0 but excluding 1. All intervals are subsets of $\mathbb{R}$ unless otherwise noted.

We use the following notation for pre-defined sets:

$\mathcal{I}$: The set of all images.

$\mathcal{A}$: The set of all affine transforms.

$\mathcal{S}$: The set of all segments observed by all views.

We begin with the assumption that in the computer vision problem, we have a set of observations from which to infer metadata.

Definition A.1 (Scene) *The scene is the four-dimensional (3D space plus 1D time) collective spatio-temporal target of the observations.*

Definition A.2 (Image) *An image is an ordered set of K-dimensional intensities located in a bounded plane.*

Definition A.3 (View) *A view $V = (I, \rho, \tau)$ is an observation of the scene, where $I \in \mathcal{I}$ is the image, $\rho \in \mathcal{P}$ is the parameters of I's projection and $\tau \in \mathcal{R}$ is the view's frame of reference.*

We then define the set of views $\mathcal{V} = \{V^n\}_{n=0}^N$, where V^0 is the scene and the planar observations of the scene are the set $\{V^n\}_{n=1}^N$.

Definition A.4 (Reference Frame) *A reference frame $r \in S$ are the parameters of a projection within a view.*

Definition A.5 (Camera) *Given a set of views $\{V^n\}_{n=1}^N$, a* camera *is a function of time where $C(t) = V^n$ such that V^n is the observation at time t.*

A.2 Segments

Segments, as we define them, are the lowest level component we actively work with.

Definition A.6 (Segment) *A* segment *is a distinct region in the image.*

Corollary A.1 (Image Representation) *Let I be any image and S be the set of all segments in I. Then $I = (S, \leq)$, an ordered set of the segments.*

Definition A.7 (Scexel) *A* scexel *(scene element) is a distinct volume in the scene which corresponds to at least one segment.*

Definition A.8 (Segment Relations) *Let S^j, S^k be the sets of all segments observed in views V^j, V^k respectively. Then the following define the relations as applied to segments:*

- Equality*: Any two segments $s^j \in S^j, s^k \in S^k$ are* equal *($s^j = s^k$) if and only if they are the same observation of the same segment such that $j = k$.*
- Inequality*: Any two segments $s^j \in S^j, s^k \in S^k$ are* inequal *($s^j \neq s^k$) if and only if they have no shared properties.*

- Equivalence*: Any two segments $s^j \in S^j, s^k \in S^k$ are* equivalent *($s^j \equiv s^k$) if they are observations of the same scexel.*

Equivalence satisfies reflexivity, symmetry and transitivity.

A.3 Correspondence

Definition A.9 (Correspondence) *Given two views V^j and V^k, and the set of observed segments in each view S^j and S^k, the* correspondence *between V^j and V^k is the bijective function:*

$$c^{j,k} : C^j \to C^k$$

where $C^j \subseteq S^j$, $C^k \subseteq S^k$, and $c^{j,k}(x) = y$ such that $x \equiv y$, $x \in C^j, y \in C^k$.

Since $c^{j,k}$ is bijective, $|C^j| = |C^k|$.

A.4 Registration

Registration: 2D affine transform between two views.

Definition A.10 (Registration) *Given two views V^j and V^k, and $C^j \subseteq S^j, C^k \subseteq S^k$ as defined in Definition A.9. Let $p : \mathcal{S} \to \mathbb{R}^2$ such that $p(x) = \vec{w}$ where $\vec{w}$ is the position of $x \in \mathcal{S}$ in the image's reference frame.*

Find the transform θ such that $\forall x \in C^j \quad \theta p(x) = p(y), y \in C^k$ and $c^{j,k}(x) = y$

where $c^{j,k}$ is as defined for correspondence.

A.5 Applied Image Registration

While the definition of image registration presented above defines the problem in a *definite* way, it does so through a purely theoretical representation which is not necessarily applicable or possible in the real world. In order to facilitate the limitations inherent in the applied representations, approximations of a number of the equations and concepts must be used. In this subsection we extend the definition of image registration to include functions and representations which allow for these approximations.

A.5.1 Applied Segments

Following directly from our representation of segment specified above, we provide two new equations which allow for the comparison of segments using a metric other than equivalency. While a theoretical function may provide a perfect mechanism for determining whether two segments from different images are equivalent, within the applied space the equivalence of segments is approximated by measuring their similarity. A segment is determined to *match* another if this similarity exceeds some threshold.

Definition A.11 (Segment Similarity) *Given two segments $s^j, s^k \in \mathcal{S}$ observed in views V^j, V^k respectively, their* similarity *is defined by the function $s : \mathcal{S} \times \mathcal{S} \rightarrow [0, 1]$ such that:*

$$\begin{aligned} s(s^j, s^k) = 0 &\Leftrightarrow s^j \neq s^k \\ s(s^j, s^k) = 1 &\Leftrightarrow s^j = s^k \\ &\Leftrightarrow j = k \end{aligned}$$

and $s(s^j, s^k) = x, x \in (0, 1)$ is a measure of the shared properties of s^j, s^k, such that larger values of x indicate more shared properties.

Definition A.12 (Applied Segment Match) *Two segments s_j, s_k are defined to* match *if their similarity $s(s_j, s_k) \geq \epsilon$, for some ϵ such that $0 < \epsilon \leq 1$.*

A.5.2 Applied Correspondence

Definition A.13 (Applied Correspondence) *Given two views V^j and V^k, and the set of observed segments in each view S^j and S^k, the* correspondence *between V^j and V^k is the bijective function:*

$$c^{j,k} : C^j \rightarrow C^k$$

where $C^j \subseteq S^j$, $C^k \subseteq S^k$, and $c^{j,k}(x) = y$ such that $s(x, y) \geq \epsilon$, $x \in C^j, y \in C^k$.

Since $c^{j,k}$ is bijective, $|C^j| = |C^k|$.

A.5.3 Applied Registration

Registration: 2D affine transform between two views.

Definition A.14 (Applied Registration) *Given two views V^j and V^k, and $C^j \subseteq S^j, C^k \subseteq S^k$ as defined in Definition A.13. Let $p : \mathcal{F} \to \mathbb{R}^2$ such that $p(x) = \vec{w}$ where $\vec{w}$ is the position of $x \in \mathcal{F}$ in the image's reference frame.*

Find $\theta \in \mathcal{A}$ such that $\sum \theta p(x) - p(y)$ is minimized $\forall x \in C^j, y \in C^k$ and $c^{j,k}(x) = y$

where $c^{j,k}$ is as defined for applied correspondence.

Appendix B

A Formal Definition of Computer Vision

One of the fundamental tenements of our interpretable model of computer vision is a formal definition. Using set theory we present here a formal model of computer vision developed by Dr. Gregor Miller in conjunction with the author, Dr. Sidney Fels, and Dr. Daeshik Jang. The undertaking of the creation of a general model of computer vision, which extends the model introduced in Chapter 4 is a challenging task and requires collaboration in order to make it feasible. Our definition of vision is introduced below, extending the pared down image registration definition to a more general definition which covers a much broader range of vision problems. Aspects of the model which were introduced in Chapter 4 are also included here in order to present the definition in its complete form.

B.1 Notation and Basic Definitions

We will use set notation to define our computer vision problems and components. A superscript denotes the view number which that item refers to, such as V^n (in this case, the nth view). A subscript denotes the item number, such as o_k (the kth object).

We use the notation $\mathbb{N}_N$ to represent the set $\{1, \ldots, N\}$ and $\mathbb{W}_N$ to represent the set $\{0, \ldots, N\}$. For intervals, we use square brackets - $[,]$ - for inclusive and round brackets for - $(,)$ - exclusive. Both may be used to define one interval e.g. $[0, 1)$ is the interval from 0 to 1 including 0 but excluding 1. All intervals are subsets of $\mathbb{R}$ unless otherwise noted.

We use the following notation for pre-defined sets:

$\mathcal{I}$: The set of all images.

$\mathcal{T}$: The interval of time across which all observations occur; the ordered set $\mathcal{T} = [t_1, t_2] \subset \mathbb{R}$ such that t_1 marks the beginning of observation and t_2 the end.

$\mathcal{A}$: The set of all affine transforms.

$\mathcal{R}$: The set of all transformations which consist of rotations and translations only (for reference frames).

$\mathcal{P}$: The set of all projective transforms.

$\mathcal{S}$: The set of all segments observed by all views over $\mathcal{T}$.

$\mathcal{E}$: The set of all scene elements (scells) observed in the scene over $\mathcal{T}$.

$\mathcal{V}$: The set of all views over $\mathcal{T}$.

We begin with the assumption that in the computer vision problem, we have a set of observations from which to infer a model or representation of the scene.

Definition B.1 (Scene) *The scene is the four-dimensional (3D space plus 1D time) collective spatio-temporal target of the observations.*

This does not limit the scene to a specific physical location, it could be many locations (as many as there are input images), and so the model may be something as simple as a classification of these, or something more sophisticated such as a geometry recovery.

Definition B.2 (Image) *An image is a set of oriented and translated regions in a bounded plane.*

Definition B.3 (View)
A view is an observation of the scene at an instant in time within $\mathcal{T}$ represented by (I, ρ, τ), where $I \in \mathcal{I}$ is the image, $\rho \in \mathcal{P}$ is the parameters of I's projection and $\tau \in \mathcal{R}$ is the view's frame of reference.

A view has a camera. A camera does not have a view given this definition (the camera is not the image)

All views in a scene are not necessarily synchronous. So we need to make assumptions on the scene or the views if we require synchronicity for a problem.

Segments are locally distinct in time, but can change over time.

Cameras could produce segments which don't change over time.

Definition B.4 (Reference Frame) *A* reference frame *provides a position and coordinate basis in the form of a transform. All local reference frames are defined with respect to a global frame, unless otherwise noted.*

The set of all reference frames is represented by $\mathcal{R}$, and used in this form in the rest of the document.

Definition B.5 (Camera) *A* camera *is a smooth continuous function* $c : \mathcal{T} \rightarrow \mathcal{V}$ *which provides the view* V *captured at time* t*.*

The views returned by a camera have the properties of being smooth and continuous i.e. the images, projection parameters and reference frame vary smoothly and continuously.

B.2 Modelling a Scene from Images

Our idea for a generic computer vision tool is to generate a model (of some kind) of a scene given a set of images as input. The definition we use for computer vision is as follows:

Definition B.6 (Vision) *Given a set of views over* $\mathcal{T}$ *observing a scene* S *and within an isolated system, the* vision *problem is to determine a model of* S*.*

The model used in this definition is hard to define globally: instead we define it for each sub-problem as the representation of the problem. For example, in image registration the model recovered is a transform which will warp the second image with respect to the first. The isolated system constraint is applied to maintain scene consistency and to define a natural boundary to what our idea of vision is. It is defined in Definition B.6, however we will need to discuss the concepts of segments and scells first to provide a concrete definition. Finally, we make no assumption on the sensor or its shape, only that it has some mapping to a planar space (although not necessarily rectangular).

We also have some guiding principles to adhere to when developing our vision framework. We constrain the work to not be subjective; this means we do not refer to vision as 'Image Understanding' or anything similar, as we are deliberately staying away from an image having any meaning. Therefore we do not allow OpenVL to maintain lists of objects or lists of names, rather we let the user deal with that themselves if they so wish, and we refer to examples or other input that has been provided. Although we cannot provide a strong constraint, we do try to be guided by the principles of determinism and repeatability in the framework. For example, given an image and asked to detect a face in that image, the computation taken should be the same

regardless of the contents of the image. And for the same image, the result should be the same across multiple executions.

B.3 Segments

One of our major goals with this work is to remove the use of pixels and frames when performing analysis on images. To accomplish this we have defined an abstraction over pixels, and attempted to provide the abstraction with properties from which we can define vision problems. The outcome of this is a representation of images which we can use to provide meaningful output to users which is also in a form useable for rendering or further processing.

Segments, as we define them, are the lowest level component we actively work with. This does not discount intensity-based or window-based methods, since the lowest-level segment could be a colour region (intensity) and windows can be defined using $([0,1] \times [0,1], aspect\,ratio)$.

Definition B.7 (Segment) *A* segment *is a distinct convex region in the image.*

Our definition of a segment forms the foundation of our approach to user-accessible computer vision. One of our goals is to eliminate the requirement of dealing with pixels and frames and instead deal with continuous, scaleable units.

Definition B.8 (Segment Convexity) *A region $R \subset \mathbb{R}^2$ is defined to be* convex *if $\forall u, v \in R$ and $\forall \theta \in [0,1]$:*

$$(1-\theta)u + \theta v \in R$$

Definition B.9 (Segment Properties) *A segment has the property of shape, and an additional finite set of properties representing important concepts of images.*

Definition B.10 (Segment Property Relations) *Here we define the relations of segments wrt properties:*

- Equality*: Any two segments u, v have* equal properties *($u \overset{p}{=} v$) if all properties (excluding shape) are exactly the same.*
- Inequality*: Any two segments u, v have* inequal properties *($u \overset{p}{\neq} v$) if and only if they have no shared properties (excluding shape).*

Corollary B.1 (Point properties) *Given the region $R \subset \mathbb{R}^2$ covered by a segment s, any point $p \in R$ has exactly the same properties (and property relations) as s. Therefore $\forall\ p \in\ R,\ p \stackrel{p}{=}\ s$.*

There may be many properties for segments, for example colour, texture, shading, shape, structure. However we do not intend to list them here, all that is important for now is to state that segments have properties, and from these we can define new operations, most importantly of which is the creation of segments given an input image (*segmentation*). The definition of segment is based on the idea of being distinct, and with properties we can now define *distinctiveness*.

Definition B.11 (Distinctiveness) *A region $R \subset \mathbb{R}^2$ is defined to be* distinct *if $\forall p_1, p_2 \in R$, $p_1 \stackrel{p}{=} p_2$ and all points adjacent to R do not have exactly the same properties.*

Corollary B.2 (Segment Overlap) *In a single image no two segments can spatially overlap.*

Proof Prove this using definition and properties of segment i.e. property equality and segment convexity.

Definition B.12 (Image Representation) *Let I be any image and S be the set of all segments in I. Then $I = (s, \tau), s \in S, \tau \in \mathcal{R}$, a set of segments where each segment has its own reference frame,with respect to the global image reference frame.*

Definition B.13 (Scell) *A* scell *(scene element) is a volume in the scene which corresponds to at least one feature.*

A scell is not necessarily distinct, even though its projection in the image may be. Distinct implies properties of segments such that the scene is not one of them.

Definition B.14 (Segment Relations) *Let S^j, S^k be the sets of all segments observed in views V^j, V^k respectively. Then the following define the relations as applied to features:*

- Equality*: Any two segments $s^j \in S^j, s^k \in S^k$ are* equal *($s^j = s^k$) if and only if they are the same observation of the same segment such that $j = k$.*

- Inequality*: Any two segments $s^j \in S^j, s^k \in S^k$ are* inequal *($s^j \neq s^k$) if and only if they have no shared properties.*
- Equivalence*: Any two segments $s^j \in S^j, s^k \in S^k$ are* equivalent *($s^j \equiv s^k$) if they are observations of the same scell.*
- Similarity*: Any two segments $s^j \in S^j, s^k \in S^k$ are* similar *($s^j \stackrel{s}{=} s^k$) if at least one property is exactly the same (excluding contour and shape).*

There is a middle ground between equality and inequality, which we will call similarity. The *level of similarity* is defined below.

Equivalence relation satisfies reflexivity, symmetry and transitivity.

Definition B.15 (Scell and Segment Equivalence) *Let S be the set of all features observed in view V. Then for some scell $o \in \mathcal{O}$, a segment $s \in S$ is* equivalent *to o ($s \equiv o$) if s is an observation of o.*

This definition also satisfies reflexivity, symmetry and transitivity, and adds scells to the equivalence class of segments (i.e. any scell and those segments which observe it are in the same class).

B.4 Objects

Definition B.16 (Object) *Let $\mathcal{F} = \{f_n\}_{n=1}^N$ be the set of all features in the scene. Then an object is defined to be the set $O = \{(f_k, \tau_k) : f_k \in \mathcal{F}, \tau_k \in \mathcal{R}\}_{k=1}^K$, $K \geq 2$, where $\tau_k \in \mathcal{R}$ defines the reference frame of f_k. The sets $O_\mathcal{F}, O_\mathcal{R}$ are the features and transforms in O, respectively.*

This definition states that an object is represented by a set of locally distinct volumes which are related by affine transforms. The simplest case is an object which consists of a single $f \in F$ and its corresponding transform is identity.

Definition B.17 (Hierarchical Object) *A* hierarchical object *is an object O with the additional constraint that $\forall f_k \in O_\mathcal{F}, k \in \mathbb{N}_K, \exists\, h : O_\mathcal{F} \rightarrow O_\mathcal{F}$ such that $h(f_k) = f_{p_k}$ ($f_{p_k} \in O_\mathcal{F}, p_k \in \mathbb{N}_K$) where f_{p_k} is the parent of f_k. Then the transform τ_k is the reference frame for f_k with respect to f_{p_k}. There is exactly one feature with associated h where $h(f) = f$ (the root).*

This definition excludes the possibility of cycles since for each f there is an h which returns a single feature; if it returned more than one parent then cycles could exist. The special case of a single cycle is dealt with by the definition that one node is a root with no parent.