2011

Weighted and filtered mutual information: A Metric for the automated creation of panoramas from views of complex scenes

Thomas Keane

WEIGHTED AND FILTERED MUTUAL INFORMATION: A METRIC FOR THE AUTOMATED CREATION OF PANORAMAS FROM VIEWS OF COMPLEX SCENES

by

Thomas P. Keane

A thesis submitted in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE
IN
ELECTRICAL AND MICROELECTRONIC ENGINEERING

Professor: _____

Thesis Advisor – Dr. Eli Saber

Professor: _____

Thesis Advisor – Dr. Harvey Rhody

Professor: _____

Thesis Advisor – Dr. Andreas Savakis

Professor: _____

Head of the Department of Electrical and Microelectronic Engineering – Dr. Sohail Dianat

Department of Electrical and Microelectronic Engineering
Kate Gleason College of Engineering
ROCHESTER INSTITUTE OF TECHNOLOGY
Rochester, New York
May 2011

DEDICATION

This work is dedicated to all of those who have stayed in my life by consistently sharing
(and by supporting me in the pursuit of)
unmitigated truth and unrelenting respect.

ACKNOWLEDGMENTS

This has been the most formative, important, and successful time of my life; and I could not have accomplished so much, or gained all that I did without RIT's wonderfully supportive and encouraging staff, professors, students, and the motivation and safety of its campus and facilities. The Department of Electrical and Microelectronic Engineering has provided amazing labs and technologies, engaged and brilliant professors, a dedicated staff, and a supportive academic curriculum that does not award anything you have not earned.

I would like to especially thank Dr. Eli Saber, Dr. Harvey Rhody, Dr. Andreas Savakis, Dr. Sohail Dianat, Dr. James Moon, and Dr. Daniel Phillips. As advisors, professors, motivators, and teachers, I could not have asked for anything better. These professors have proven time and again that a fast-paced curriculum, a robust exploration of topics, and all the time constraints and pressures for success could not be overcome without the guidance and support of such a knowledgeable, dedicated, welcoming, encouraging, and engaging set of individuals.

I would like to thank and acknowledge Michael Regelski, Stephen DeBellis, and Jeffrey Raj from Lenel Systems, Inc. a UTC Fire & Safety Corporation for their support, funding, and dedication throughout this project.

I would like to also acknowledge Michael Pecoraro, David Wagner, Haleem Syed, Sreenath Rao Vantaram, Xiaofeng Fan, Mustafa Jaber, and all my peers and friends in the Department of Electrical and Microelectronic Engineering and the Chester F. Carlson Center for Imaging Science. Having such a great group of peers and friends has not only made this experience so much more successful, but without their advice and our shared interests and dedication, I am certain that I would not have made it to where I am today.

Lastly I would like to especially acknowledge those who have loved and supported me: Rachel Chrash, AJ Connors, Siddharth Khullar, Abby McCarthy, Susan Pedrotti, Antonella Bonfanti, Blake Huang, and all the rest of my family and friends.

CONTENTS

Abstract . . . . . **vii**

1 Introduction **1**
1.1 Objectives and Motivation 2
1.2 Literature Review 3
1.3 Contributions 7
1.4 Applications 8
1.5 Implementation 10
1.6 Thesis Outline 11

2 Background **13**
2.1 Digital Image Generation 13
2.2 Histograms and Color Imagery 16
2.3 Mutual Information 19
2.4 Camera Geometry 23

3 Algorithm and Implementation **28**
3.1 Algorithm Components 28
3.2 Feature Extraction 30
3.3 Affine Transform Search 33
3.4 Weighted and Filtered Mutual Information 36
3.5 Stitching and Blending 43

4 Results **49**
4.1 Affine Views 50
4.2 Near-Affine Views 59
4.3 Complex Views 64

5 Conclusions and Future Work **70**
5.1 Future Work 70
5.2 Final Remarks 72

References **73**

LIST OF FIGURES

1.1 Projective Camera System Model 2

2.1 The Bayer pattern image resultant from the bayer filter capture. 14
2.2 Descriptive diagram of the demosaicking to create the RGB channels from the Bayer pattern image. 14
2.3 Venn diagram of key measures in Information Theory 20
2.4 Terminology and structure of our application. 24
2.5 Projective Camera System Model 25
2.6 Multiple Projective Cameras Model 26

3.1 Algorithm Flowchart 29
3.2 Example Gradient Map Histogram with Quantization Boundaries 32
3.3 Depiction of Images as Layers in single Image Coordinate Space 43
3.4 Rooftop Scene (a) Left View Padded, (b) Right View Padded, (c) Overlapping Padded and Transformed Views 46
3.5 Laplacian Pyramid Blending Diagram 47

4.1 Rooftop Views (a) Left View, (b) Right View 53
4.2 Rooftop Views Blended 53
4.3 Rooftop Views Blended Manually (Affine) 54
4.4 Mutual Information Maps from Translation Search (a) Filtered and Weighted, (b) Weighted 54
4.5 Rooftop Views with Average SNR of 24.122 dB (a) Left View, (b) Right View . . . 55
4.6 Rooftop Noisy Views Blended (Average SNR: 24.122 dB) 55
4.7 Stone Wall Scene (a) Left View, (b) Right View 56
4.8 Stone Wall Views Blended 56

4.9 Mutual Information Maps from Translation Search (a) Filtered and Weighted, (b) Weighted 57
4.10 Stone Wall Views with Average SNR of 16.473 dB (a) Left View, (b) Right View . 57
4.11 Stone Wall Noisy Views Blended (Average SNR: 16.473 dB) 58
4.12 Stone Wall Views Blended with 1% Overlap 58
4.13 Art Gallery Scene (a) Left View, (b) Right View 60
4.14 Art Gallery Views Blended 60
4.15 Art Gallery Views Blended Manually (Affine) 61
4.16 Art Gallery Scene (Modest Angle) Views (a) Left View, (b) Right View 61
4.17 Art Gallery (Modest Angle) Views Blended 62
4.18 Art Gallery (Large Angle) Views (a) Left View, (b) Right View 62
4.19 Art Gallery (Large Angle) Views Blended 63
4.20 Art Gallery Scene 01 (a) Left View, (b) Right View 66
4.21 Art Gallery 01 Views Blended 66
4.22 Art Gallery Scene 02 (a) Left View, (b) Right View 67
4.23 Art Gallery 02 Views Blended 67
4.24 Lenel Front Lot Scene (a) Left View, (b) Right View 68
4.25 Lenel Front Lot Views Blended 68
4.26 Lenel Back Lot Scene (a) Left View, (b) Right View 69
4.27 Lenel Back Lot Views Blended 69

ABSTRACT

To contribute a novel approach in the field of image registration and panorama creation, this algorithm foregoes any scene knowledge, requiring only modest scene overlap and an acceptable amount of entropy within each overlapping view. The weighted and filtered mutual information (WFMI) algorithm has been developed for multiple stationary, color, surveillance video camera views and relies on color gradients for feature correspondence. This is a novel extension of well-established maximization of mutual information (MMI) algorithms. Where MMI algorithms are typically applied to high altitude photography and medical imaging (scenes with relatively simple shapes and affine relationships between views), the WFMI algorithm has been designed for scenes with occluded objects and significant parallax variation between non-affine related views. Despite these typically non-affine surveillance scenarios, searching in the affine space for a homography is a practical assumption that provides computational efficiency and accurate results, even with complex scene views. The WFMI algorithm can perfectly register affine views, performs exceptionally well with near-affine related views, and in complex scene views (well beyond affine constraints) the WFMI algorithm provides an accurate estimate of the overlap regions between the views. The WFMI algorithm uses simple calculations (vector field color gradient, Laplacian filtering, and feature histograms) to generate the WFMI metric and provide the optimal affine relationship. This algorithm is unique when compared to typical MMI algorithms and modern registration algorithms because it avoids almost all *a priori* knowledge and calculations, while still providing an accurate or useful estimate for realistic scenes. With mutual information weighting and the Laplacian filtering operation, the WFMI algorithm overcomes the failures of typical MMI algorithms in scenes where complex or occluded shapes do not provide sufficiently large peaks in the mutual information maps to determine the overlap region. This work has currently been applied to individual video frames and it will be shown that future work could easily extend the algorithm into utilizing motion information or temporal frame registrations to enhance scenes with smaller overlap regions, lower entropy, or even more significant parallax and occlusion variations between views.

CHAPTER 1: INTRODUCTION

Before delving into the depths the algorithm, its development, its implications, its contributions, its results, and its potential future, there needs to be an introduction to what has been, and what is trying to be, achieved here. Mathematics and programming are languages, they are functional, and they are tools to serve a purpose. Without understanding that purpose, without preparation and a solid foundation, there would be nothing but a set of facts and figures susceptible to misinterpretation. The core of the WFMI algorithm is in its implementation through a thorough understanding of the underlying mathematics and practical considerations. A lot of the following discussion is then focused on explaining where the application of these operations follows from, and why and when it succeeds or fails. In order to maintain a clear and consistent understanding the following terms and phrasings will be used.

A **scene** is understood here as the real-world location being imaged, and the camera imaging the scene is known as the **view** (of the scene). The output of the camera is understood to be a digital tri-chromatic video composed of tri-chromatic (typically RGB) **frames**. The most important term will be the **overlap** between views. In this context we are defining **overlap** to mean the regions within the views' frames which have imaged the *same existing region* within the scene. Characteristics of imaging geometry will not always allow for identical overlap regions in the frames, but our definition will apply to the understanding that there is real-world correspondence between the objects from the scene imaged onto the frames through the multiple views.

In order to maintain consistent terminology across the development and implementation, MATLAB® notation for image sizing will be followed, where each image is of size $\mathfrak{m} \times \mathfrak{n} \times \mathfrak{p}$. The row, column, and channel indices will come from the following closed sets, respectively: $[1, \mathfrak{m}]$, $[1, \mathfrak{n}]$, and $[1, \mathfrak{p}]$. While it is more mathematically apt to model each frame (image) from each view as the realization of a random process (the camera capturing the scene), we simplify the terminology and understanding to assume each frame is a discrete random variable that is a 3-Dimensional (3-D) set of b-bit intensity values. To characterize these random variables (R.V.) we develop a model of

their 1-D Probability Mass Function (PMF) based on features (a 2-D array) from the images. Each image's PMF is then an approximation of the characterization of the real-world scene; which under a continuous R.V. model would be its Probability Density Function (PDF). It is understood, and applied in development, that each image has come from a projection, sampling, and quantization of the scene at that view, as shown in part in Figure 2.5.

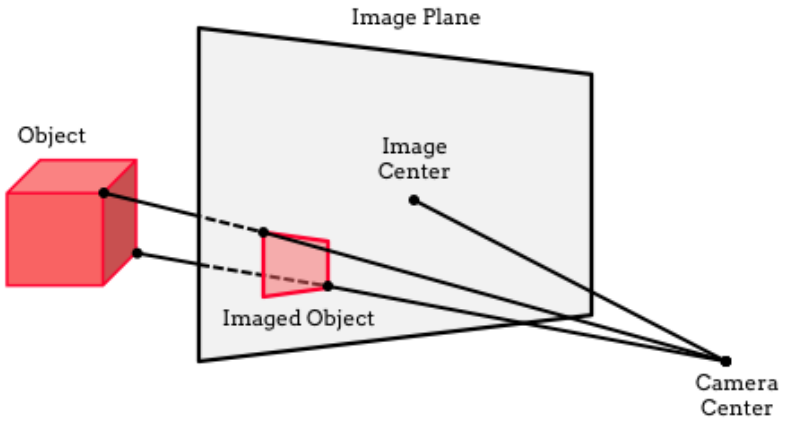


Figure 1.1: Projective Camera System Model

1.1 Objectives and Motivation

Image registration is such a rich topic of interest because in the currently available algorithms there are often more constraints than are practically allowable or desirable. For modern security surveillance scenes, one camera will never be enough to sufficiently monitor the regions of interest in a scene. It is also often a waste of time and money to have a technician or operator generate *a priori* point correspondences, even though it would allow for extremely accurate registration, as will be discussed in the next section. Single-view time-lapsed video mosaicking provides very little practical purpose in the surveillance field since there is no temporal registration (only spatial) and the larger the scene the more temporal variation there will be across the data set. It is immediately clear that in order to identify, track, or monitor individuals or areas of interest, as is the goal of security and surveillance systems, sufficient coverage and temporally concurrent views are

essential. Multi-view video registration would provide the two.

Contemporary panorama creation research is extensively studied in situations of large sets of input images with large amounts of redundant overlaps, and views are often taken from a central viewpoint or a single moving camera. There also advanced algorithms relying on the complex mathematics and numerical methods techniques in the development of registering sparse matrices. These are very important and interesting areas of research, but they are respectively too impractical or too complex for generating a simple, accurate, and automatic algorithm. Surveillance systems are designed with a focus on cost-benefit analysis, which tends towards utilizing fewer cameras with minimal overlap in order to cover the most surveyable area while maintaining continuous spatial and temporal observation. This is the main motivation for this algorithm, to look at a realistic scenario, the surveillance of complex real-world scenes, and provide an automatic, fast algorithm to stitch together overlapping views with no prior knowledge of camera relationships which will present a convincing view of the observed scene.

By understanding the desired scenarios, as will be discussed in Chapter 3, it was found that a standard MMI approach is significantly susceptible to false positives in realistic surveillance scenes. The goals of this research were to provide convincing views, fast and frugal processing, and a versatile and extensible algorithm. The academic and scientific pursuits of the completion of this thesis pushed for consistent theoretical development, while the motivation for the research was to meet the project goals and provide a useable algorithm. In order to succeed it was necessary to develop an in-depth understanding of the practical considerations that ultimately guided the applied theoretical development. The WFMI algorithm is a prime example of modern research on the implementation of maximized mutual information (MMI) based registration algorithms with a novel approach to the complexities that arise in realistic surveillance scenes.

1.2 Literature Review

Contemporary work in image registration has progressed from basic applications of projective geometry theory to solving large-scale and complex problems. Research continues more and more

in applications that extend the limits of the theoretical developments. The work presented here has looked at multiple views of complex scenes susceptible to parallax and object occlusion disparities, as well as the standard concerns of depth variation and the complexities of projective geometry within multiple views. Looking through the current research presents what has been done, what is developing in the field, and where and how the WFMI algorithm fits in with modern research.

An excellent survey of the field of image registration was done by the Zitová and Flusser in [1]. Their work very succinctly described the four major steps of most modern registration algorithms: feature detection, feature matching, transform calculation, and image transformation. This is an excellent distillation of all major techniques, and in Chapter 3 the WFMI algorithm's approach will be compared to this standard methodology. A lot of the research in the field is focused on the first two of the four steps; feature detection and matching. In these areas of research a new method appears rarely and most techniques are based on a few general methods that are refined as research progresses (cross-correlation, mutual information, Fourier methods, corners and edges, gradient descent algorithms, *etc.*). Foundational work that is still in very wide use today was done by [2] and [3] in looking for an affine invariant set of features and a method for their correspondence.

There is also another well-written survey of medical image registration, and more specific to applying mutual information techniques [4], from the same year (2003). Again, many of the techniques rely on only a few general algorithms that are then modified, aiming for refinement, with new measures or weighting schemes and various optimization techniques. The work in [5] was foundational in mutual information techniques and provides a very in-depth understanding of the derivation of mutual information as a correspondence metric. Looking through [1] and [4] can give an excellent first exposure to image registration and the application of mutual information, but being over 8 years old at the time of this writing there are more contemporary techniques and research being applied.

More recent research such as [6], [7], [8], [9], [10], [11], [12], and [13] have all advanced the field greatly by exploring new techniques and new applications for the general methods found in [1] and [4]. What is still of peak interest, though, is the feature detection and feature matching

stages of general image registration. Homography calculation and image transformation are extremely well-theorized by the mathematics of projective geometry, but determining a homography through feature detection and feature matching seems to require more insight into the cognitive understanding of imagery.

The work in [2] is an excellent example of the application of David Lowe's foundational SIFT algorithm. SIFT stands for Scale Invariant Feature Transform, and is widely used in research for the feature detection and matching stages. Especially as research has progressed into recognizing projective geometry concerns and more complex multi-view scenarios, it is becoming more and more crucial to define invariant features, as opposed to the previous industry standard of using corner features as developed in [14]. While Lowe's work does not provide the perfect invariant features, it is still the research standard, although it is a patented algorithm requiring a commercial license. Work such as [9] and [12] are basic examples of the continued research on the SIFT operation. And while it is a great tool, the proprietary nature was incompatible with the grant-funded research done here. It is also a complex technique that only provides the features and estimated matches, it does not explicitly or uniquely correspond the features, often requiring the implementation of a RANSAC algorithm [12]. This pushed our research away from a similar technique as it seemed unlikely to move towards an efficient and simple real-time implementation for color video surveillance scenarios.

The work in [6] and [8] takes into consideration projective geometry, stereo image processing, and multi-view concerns in order to develop 3-D scene models. This was inspirational and motivational work for the WFMI algorithm as it presents the rich amount of data and information available in registered imagery. As will be discussed further in Chapter 5, the WFMI algorithm's success is applicable to future work in depth reconstruction, scene understanding, and the potential for becoming an iterative algorithm that could be self-improving in determining the precise view-to-view homography. Stereo correspondence provides an immense amount of information about scene content and geometries, making it a quickly emerging topic of interest.

In terms of feature correspondence, RANSAC [12] is still the most popular technique, while

supervised correspondence continues to be in widespread use as well. The problem of feature correspondence is extremely difficult as it requires some knowledge or assumption of the structure of the views or the scenes once features have been identified. Once the features can be corresponded, the actual registration of the images and the homography generation are extremely well defined by projective geometry as detailed extensively in [15] and [16]. Leaving the only other major concern to be the actual stitching and display of the registered images.

Again in [1] stitching techniques are described, and there are novel techniques still being developed as in [10]. The difficulty in stitching is that its accuracy is often entirely dependent upon the accuracy of the homography. Given that the WFMI algorithm can only assure an accurate estimate in general cases, most stitching techniques prove to be ill-posed as it is known that the transformed pixels are not perfectly corresponded in all results for the WFMI algorithm. However, the research done in [17] developed a Laplacian pyramid blending technique that blends frequency content of images without requiring any spatial correspondence. In [17] images are shown that have been blended for artistic effect, an apple and orange for example, which have not even been through a registration algorithm. Chapter 4 will discuss the results in detail, but the multi-resolution spline blending technique is vastly superior to any other color correction, stitching, or blending algorithm for this implementation because perfect correspondence accuracy is not required to create the convincing view. Ideally there should be a utilization of the technique in [10] combined with the work from [17], because then the location of the view transition (the stitching seam) could be hidden; yet since the WFMI algorithm is not providing complete accuracy in the general projective case it was determined as a matter of future research to improve the pixel-based correspondence accuracy of the panorama. Again this will be discussed in further detail in Chapter 5.

The WFMI algorithm development took careful consideration of previous work, especially as highlighted by the research surveys in [1] and [4]. Yet the WFMI algorithm was tasked to overcome a very open problem: fully automatic registration, unknown camera locations, no camera calibration, and to produce a convincing panoramic view. These conditions are far more open-ended than most techniques in practice, and any work facing similar conditions were often solved

through the use of SIFT and RANSAC, or their derivatives. Or there were the large-scale computation problems, such as the excellent investigation in [18] using public image databases for synthesized scene tourism. The WFMI algorithm has followed a different path from most of the modern research, but is deeply rooted in many shared theories, making it extremely important to understand the current work and their limitations and goals. The discussion presented here should also provide added insight into strengths and limitations of multi-view registration, and where the field of research has yet to extend.

1.3 Contributions

The WFMI algorithm is a novel set of means to perform unsupervised, automated panorama creation for surveillance, and other realistic, scenes. The results of the algorithm are convincing, blended views generated by the affine homography derived from the affine transform search between the scenes. This is a robust and novel affine registration algorithm that can be applied to real scene views to provide useful estimations in registering these views of scenes that well beyond the constraints of an affine relationship. The initial implementation of an MMI algorithm gave light to the necessity of incorporating practical considerations in order to successfully utilize mutual information as a robust correspondence metric. The novel weighting and filtering aspects of the WFMI algorithm go beyond a simple normalization process; these steps are actually crucial to getting accurately registered views from realistic scenes or difficult affine scenarios, especially in scenes with lower entropy or minimal amounts of overlap.

There is no camera calibration step, there are no initial correspondences, and there is no camera location or orientation assumption. Obviously if there is no amount of entropy in the views, or no actual overlap, any algorithm would fail. In terms of the WFMI algorithm it was found, as was expected, that minimum tolerable overlap between the views is inversely proportional to the amount of entropy in the overlap regions and in the scenes themselves. These were empirically discovered and while there are no hard-and-fast limits presented here, affine scenes could tolerate down to 1% overlap while realistic views could tolerate down to 10% overlap.

By developing a strong understanding of the principles of image registration, this algorithm shows that typical constraints and models can be expanded both mathematically and in practice. Known camera parameters, camera spatial relationships, and *a priori* point correspondences have been clearly shown to be significant aids to getting from two disparate images to a registered panoramic view; but the WFMI algorithm shows that these are not the only means to do so, especially when re-incorporating an understanding of the foundational mathematics and physics of the views and scenes. Practical considerations motivated the generalization of the algorithm, which push the limits of contemporary research in automated image registration, especially for MMI algorithms. The WFMI algorithm contributes a novel extension of mutual information based correspondence methods through application to complex scenarios with illumination variations, parallax disparity, and occlusion disparity between multiple views.

While an affine homography is derived for certainly non-affine views, it is an accurate estimate and can very easily be used as a robust and intelligent set of initial conditions for point-to-point correspondence algorithms. This contribution of the WFMI algorithm could greatly reduce the mathematical and algorithmic complexity of the stronger techniques in the field of research, given that knowledge of the amount and general region of overlap between the views is also considered known in contemporary techniques. This could be used in a realtime system where the WFMI algorithm runs initially and is updated by a very simple feature-to-feature search in a small neighborhood, as it will be accurately assumed that corresponding features are nearby in the transformed view(s) presented by the WFMI algorithm.

1.4 Applications

This algorithm has been developed for indoor and outdoor security surveillance purposes, but holds a lot of more general theoretical and practical weight. In security applications the goal is to avoid equipment costs and confusions, specifically in viewing multiple scenes simultaneously. A panorama can avoid the use of those multiple monitors or multiple windows, thus allowing for a whole area to be viewed not only simultaneously but also contiguously. A blended panorama

created from multiple-views with parallax and occlusion artifacts closely follows our own human visual system's means of perceiving the world [19], and so these are by no means preventative to creating a convincing view. Thus by creating a contiguous, coherent, and convincing view, the users and operators can eliminate one more step in processing the scene(s) and extracting the useful information. This is not only useful for watching surveillance videos, but also for storing and processing important scenes; for which this algorithm provides substantial, foundational improvements that lend themselves to other areas of image and scene processing.

A multi-view tracking algorithm becomes a single-view problem when all views are registered. And any motion-based algorithm is provided with more causal data, creating a robust initial data state for any type of memory-based framework, as modern video algorithms tend to be computationally exhausting. Surveillance systems often implement algorithms for spatial detection, awareness, and warnings, such as a watchdog system that provides an alert when objects enter a secured area. Systems such as these are limited by their views, and by the data provided. A spatially contiguous and temporally coherent panorama of a scene involving a series of distinct camera views with minimum overlap can allay that concern by expanding the input data set for automated scene description and analysis algorithms. Panoramas, in this context, are expanded data sets. Especially considering that video data contains spatio-temporal information regarding scenes, and without accurate spatial and temporal relationships the conglomeration of multiple videos to constitute a data set becomes a hindrance to generating or applying accurate tracking and motion algorithms. The results of applying the WFMI algorithm to scenes with parallax and occlusion will produce useful results, especially for objects in motion as they pass from one view through the overlap and into the other. Mosaicking will fail in surveillance motion or tracking algorithms because it provides no temporal coherence; and while overly redundant data sets (large overlaps from many cameras) could simplify this process, they are a front-end and continued maintenance cost added to the implementation. Pixel-to-pixel registration algorithms in complex scenarios can often be too theoretically restricted to provide useful results, especially in object tracking when occlusion is present.

The WFMI algorithm is best lent to an application where modern means of registration and panorama creation are too academically or scientifically stringent in their conditions, too computationally complex, or too cost prohibitive in the required hardware or software licensing. Surveillance was the intended application for this algorithm, but the following discussion of the implementation will show that any system with minimal *a priori* scene knowledge and a need for unsupervised automation could find a solution here.

1.5 Implementation

The WFMI algorithm was written initially in MATLAB®; then it was ported over for Lenel Systems, Inc. (the grant provider) to an implementation in C++ through the use of the OpenCV library. The MATLAB® development environment proved an excellent prototyping system that allowed exploration and testing concurrent with theoretical development. This did result in longer run-times and tended towards minor computational bloat, but during that stage of the research the goal was proof-of-concept, while providing a path for future work that could eventually move towards real-time processing. However, several optimizations were made to continue on schedule, such as the use of a hierarchical search and vectorization of the histogram calculations. Since this project was funded through a corporate grant, it was designed for corporate review/preview and was ultimately implemented into the grant provider's system to be sold as part of their product. This pushed the design towards their system's constraints and their user base's practical needs, while maintaining a robust scientific and theoretical foundation. As previously mentioned, this algorithm has been designed for surveillance, but its concepts are more widely applicable in the field of automated image registration.

The MATLAB® implementation went through several initial rewrites, and the first stage of development was dedicated to a MATLAB® based computationally efficient algorithm. For example, the translation and mutual information search portions of the algorithm were cut down from an initial computational time of 2 hours to less than a minute. This was motivated by scheduling and concerns with developing the theory in concurrence with testing, but also provided significant

insight into the theoretical development. The OpenCV implementation was not optimized for C++ at the time of this research, but did take advantage of some of the algorithmic enhancements that were applied in the MATLAB® prototyping development.

It should be noted that the MATLAB® and OpenCV implementations truly complemented each other in providing a deeper understanding that produced the final algorithm and the theory to be discussed here. After having a working algorithm in MATLAB®, the development process for the translation to OpenCV provided significant scientific and theoretical insight into the limitations and abilities of the algorithm and its foundations given the strict memory and type constraints in C++ compared to MATLAB®. The difference in the quantization and strict-typing of data in the C++ implementation showed the limits of the entropy based measure and provided insight allowing for the reorganization of the algorithm workflow, so as to "carry along" as much of the original views' information until reaching the correspondence stage.

1.6 Thesis Outline

Chapter 2 will present the requisite background information. While some basics of digital image sampling, information theory, and probability theory will be assumed known. The second chapter is presented as the essential mathematical and conceptual theories required to understand the algorithm and its development. This includes a conceptual understanding of information theory and histogram generation, as well as a mathematical development focusing on probabilistic measures. Another major component of the algorithm is digital filtering and this area will be generally assumed to be understood as it is a very small component and will be discussed in the implementation. This chapter will also cover digital interpolation and decimation, image formation, digital sampling, digital color images, and digital pattern recognition theory. This chapter will push towards a higher level of understanding based on well-detailed theories mathematics referenced in texts and publications. Chapter 2 will provide the theory needed and developed in taking the mathematical details and expanding them to the practical scenarios and application of the algorithm.

Within Chapter 3 will be the algorithm itself. As Chapter 2 is providing background mathematics, Chapter 3 is the application of the presented theories and concepts to describe the completed algorithm. Chapter 3 is heavily geared towards understanding the choices made in the inception and prototyping of the algorithm that lead to the current state of the algorithm. This information provides not only a very detailed understanding of what was done and achieved, but illuminates the necessary details required in judging the success of the algorithm. Many changes were made throughout the development of this algorithm, especially given that there was a substantial conceptual and mathematical divergence from standard MMI algorithms.

Chapter 4 will illuminate the results: how they were achieved, what they show, and what possibilities there are for further development to these examples. What will be presented in this chapter is not just a display of results, but also a discussion of what the algorithm produces and why. The conditions of the image capture and formation are the major factors in the variability of the results.

Finally, with Chapter 5, this paper looks back to what has been developed and forward to what extensions and applications apply. A summary of the algorithmic, scientific, practical, and academic results is presented. More detailed applications and possibilities are highlighted, given that the whole of the algorithm has since been presented. And ultimately a look towards the future is provided. Not just future work on the algorithm itself, but the future advancements in the field of image registration that could possibly branch off from what was found in this research.

CHAPTER 2: BACKGROUND

Before delving into the details of the algorithm and its implementation, it is first necessary to understand some basic and complex mathematical concepts from the fields of random signal theory, information theory, and digital image and video processing. With a robust exploration of the conceptual results of these mathematical and scientific topics, the practical considerations that molded the development of the algorithm will become clear. Even with a deep understanding of the theories to be refreshed here, it was the refracted view of these principles through the lens of the requisite practical considerations that made the algorithm successful.

2.1 Digital Image Generation

First it is necessary to understand digital image systems from a statistical viewpoint; from the capture stage through to the processing stage. Initial digital capture is through the transformation of incident photon energy that is thresholded into digital signals describing what can be referred to as intensity. To capture color information in a typical CCD array (single channel) the Bayer mask is the most prevalent methodology. It is based on the trichromatic theory of color from the scientific understanding of the human visual system [19], by applying the known sensitivity of the human visual system to middle range wavelengths of visible light (green light). Therefore in the Bayer mask there are twice as many green (G) samples as red (R) and blue (B). What is important to note is that in generating the final color image, there is an application of an intensity interpolation method, known as demosaicking. Each pixel, representing a point in image space (the view), that corresponds to a point in the world space (the scene), is only sampling one color (wavelength) of light: R, G, or B. So for a color image the color values in all the other pixels are interpolations of neighboring pixels. See Figures 2.1 and 2.2 for a visual description of this process.

Demosaicking is a well-understood and well-researched topic of interest where theories are being developed and applied to improve the interpolation results. The point to grasp here is that

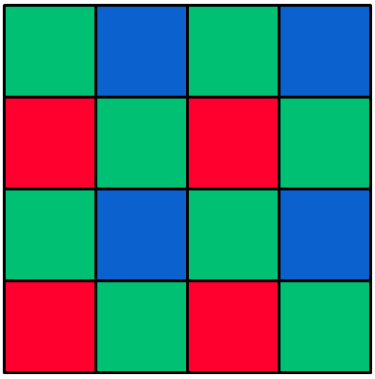


Figure 2.1: The Bayer pattern image resultant from the bayer filter capture.

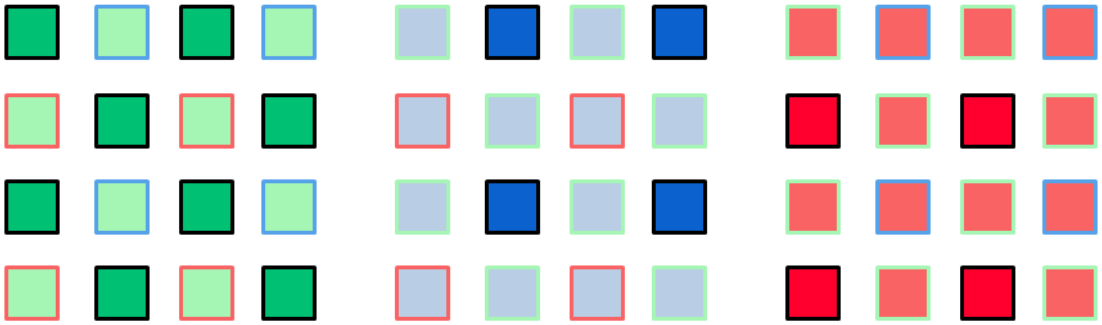


Figure 2.2: Descriptive diagram of the demosaicking to create the RGB channels from the Bayer pattern image.

color imagery, even before any image processing algorithms are applied, is a sampled, quantized, and interpolated array of data. This shows a significant limitation in the accuracy of the initial information within a digital color image. While by no means a crippling factor, understanding this process allows for developing more scientifically stringent arguments for the concessions made later on in the development of the algorithm as further quantizations and interpolations occur.

Parallax and occlusion variations will occur with views at different spatial locations, but this also shows that intensity and color variations will occur in spite of any spatial variation. The views are inherently non-identical in terms of intensity/color, even when capturing the exact same object at the exact same depth and angle, from the same exact camera at different times. Understanding the innate statistical variation present in a digital image is carried throughout this discussion.

Therefore, at this stage, a digital color image is seen initially as the demosaicked (interpolated)

result of the Bayer pattern image. Stemming from that, a digital image is understood as a rigid, rectangular array of variable bit and channel depth. The number of channels pertains to the color range and the bit depth pertains to the intensity range. Standard contemporary color images are 3-channel (RGB) arrays of the notational size: $\mathfrak{m} \times \mathfrak{n} \times \mathfrak{p}$ (rows-by-columns-by-channels), and have a per pixel bit-depth of 8-bits, resulting in the descriptor of a 24-bit (3-channels with 8-bits each) color image. Variations of these characteristics exist in widespread use, and are important to understand, but the fundamental developments being made here can be easily translated to other image color-types and bit depths, thus all images in the rest of this work will be assumed 8 bits-per-pixel (bpp), 3-channel RGB images. But, the most fundamentally important concept in order to maintain a successful grasp of image processing is the notion that a digital image of size $\mathfrak{m} \times \mathfrak{n} \times \mathfrak{p}$ is akin to a $\mathfrak{p}$-dimensional discrete-space random variable. More generally a digital image should be viewed as a realization of a random process, but the concept will remain apt by discussing images as random variables and it will avoided unnecessary mathematical complexity in this conceptual presentation.

The discussion will be simplified to the notion of an $\mathfrak{m} \times \mathfrak{n} \times 1$, henceforth $\mathfrak{m} \times \mathfrak{n}$, image. The digital image (a frame from the view of the scene) is known as a projected, sampled, and quantized version of the theoretical real world proposed model of the scene. The approximating function's discrete result over the image domain, referred to for simplicity as the intensity image (a random variable), is thus a function of the set of discrete random outcomes of the image capture (the experiment). These terms follow from [20] where the capturing of the light reflected off the scene is understood as the **experiment**. Each pixel is the projected, sampled, and quantized intensity of that light, and is understood as the **outcome** of the **experiment**. Thus the image that is formed can be seen as a function on those intensities and is therefore known as the **random variable**. The captured image is the two-dimensional (2-D) ($\mathfrak{m} \times \mathfrak{n}$) discrete random variable that is the projected, sampled, and quantized version of the continuous random variable that describes the scene.

The real motivation of these notions is that, as random variables, the images of a scene can each have their own Probability Mass Function (PMF) [20] which characterizes them. In our discussion

we are making the assertion that the Probability Density Function (PDF) that describes the model random variable of the scene is being approximated by the PMF of the image that captures that scene. Each distinct view of the scene should be modeled by a distinct function of the intensities reflected and refracted by the objects in the scene, thus a distinct continuous random variable that will be approximated through image capture by the discrete random variable (image) associated with that view. And so the overarching fundamental assertion being made is that overlapping views should share the same scene model in the overlap. And so a joint PMF between two overlapping views will be non-trivial and non-separable, asserting that distinct overlapping views are not independent because they are modeling the same scene PDF, through their approximations. This is the foundation for the use of the mutual information metric. Non-mutually exclusive random variables will have some non-trivial amount of mutual information between them.

And so the discussion is focused on the images captured from the scene. The scene view has the PDF $\tilde{p}_i$ while the image from the view has the PMF p_i. The rest of the algorithm is built on the registration of two frames, extensible to multiple frames, and so the two images will be described more simply as random variables A and B. Thus they will have a joint probability mass of $p_{AB}(\mathbf{a}, \mathbf{b})$ and their marginal probability masses $p_A(\mathbf{a})$ and $p_B(\mathbf{b})$, where $\mathbf{a}$ and $\mathbf{b}$ are the outcomes (the vectors of intensities or feature values at the respective pixel locations). The 1-D PMF for digital images is accepted in practice as the normalized intensity histogram, and the 2-D Joint PMF is the normalized bi-variate intensity histogram. The next section will discuss the relationship between the image histogram and the proposed PMF notation, which will be used to characterize an image (the random variable) and its relationship to the corresponding scene.

2.2 Histograms and Color Imagery

A digital image histogram [21], referred to by $\mathbf{h}(\mathbf{n})$, is a discrete function (a vector) that represents intensity value totals (counted) from the related digital image, and whose independent variable ($\mathbf{n}$, a vector) represents the center value of the subset of intensities from the related digital image, called the **bin center**. In this discussion we are using integer width bins on integer data,

so each bin will be described by the interval in Eq. 2.1. And so an intensity histogram is a vector holding the total number of intensity values in the intervals related to the indices of that vector.

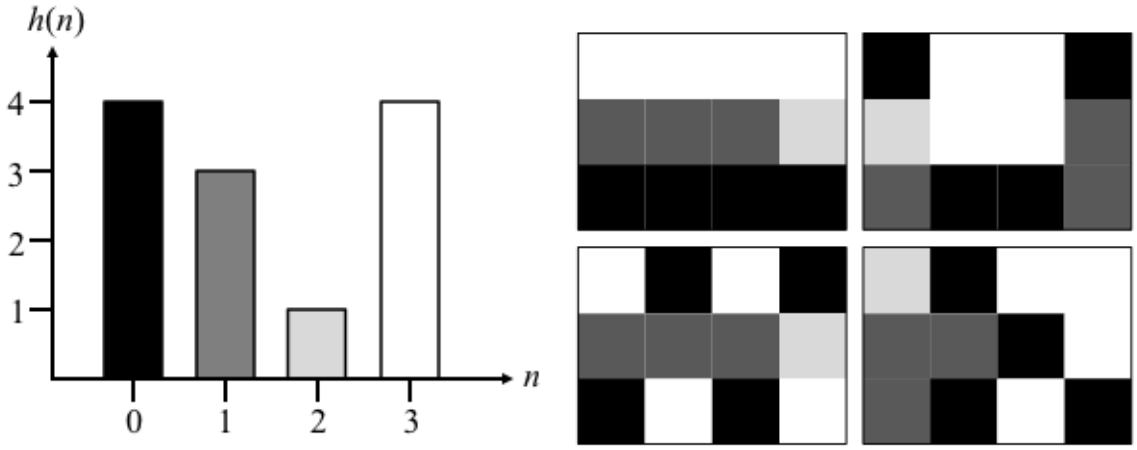
$$[n_i, n_i + 1) \tag{2.1}$$

For 8-bits per pixel (8-bpp) single channel images, this allows for a maximum of 256 unique intensity values per pixel. And again these intensity values are assumed to be statistically random over the 2-D spatial domain of the image. The total sum of that histogram, as shown in Eq. 2.2, is the product of the image dimensions, $\mathfrak{m} \cdot \mathfrak{n}$, provided the bins cover the full intensity range as developed here.

$$\mathfrak{m} \cdot \mathfrak{n} = \sum_{i=0}^{255} h(n_i) \tag{2.2}$$

Any digital image histogram that covers the entire image and the entire intensity range will always sum to the image size, because all intensity values will fall in the range of the histogram bin intervals, by construction, and can only be counted once, by definition. This is how the PMF can be constructed; by normalizing a histogram by its sum total. The normalization will impose the constraint that an image's intensity histogram's sum will equal 1, which is a condition of a PMF [20]; and clearly the histogram describes how many pixels correspond to an intensity value or range, which when normalized can be seen as a corollary to the probability (or frequency) of that intensity value or range of intensities occurring within the image. The specificity of the histogram to the image must be recognized though; because what the histogram conveys is that the pixels in *this* image are distributed in *these* amounts. The imaging equivalence made here is that the normalized histogram is conceptually generalized to be a descriptor of the image, and it is also the approximation of the descriptor of the real-world scene that that image has captured by its camera's view.

After understanding what a histogram is and how to build it, its use is only as good as its limitations. And one of the most important limitations of an intensity histogram is the loss of spatial



(a) Example Intensity Histogram

(b) Example 2-bpp Images with Identical Intensity Histograms (0 to 3 :: Black to White)

information in its generation. No spatial information is mathematically present in the calculation of a basic intensity histogram, as it has been described and used herein. To illustrate this concept Figure 2.3(a) displays a very simple histogram for a 2-bpp image of size 3×4. Figure 2.3(b) shows 4 unique images that all share the histogram shown in Figure 2.3(a). These images come from the set of 138,600 possible images $\left(\frac{12!}{4!\cdot 3!\cdot 1!\cdot 4!}\right)$ that share the histogram in Figure 2.3(a). But this is just a purely mathematical example with no constraints. By the nature of reality, such as the spatial and temporal continuity of objects within reality, images of real scenes are highly constrained subset of all the possible images from the total set (all the possible combinations of intensities given the number of pixels in the image). Any 2-bpp image of size 3×4 is actually in the general set of 4^{12} images, and so the set of images with the histogram in Figure 2.3(b) is roughly less than 0.8% of the total set, while still unconstrained to real world shapes. Extending this idea to the images used in this algorithm, which were all roughly of the size 480×640 and were 3 channel images of 8-bpp, that gives 768 possible values per pixel location ($3 \cdot 2^8$). This means that there are $768^{480 \cdot 640}$ possible images of this size with no spatial or histogram constraint. Any histogram, color constancy model, texture, object contiguity, or any other real world aspect or feature relevant to the digital image will create a smaller and smaller subset. It is thus applicable to determine that the approximation of using a normalized histogram to produce the PMF of an image is a theoretically relevant and an apt,

realistic association. Conceptually what is being stated or assumed is that when viewing the same scene, two different views will be sampling the same random function, meaning that they should be describable by the same PMF in the overlap region, as that scene is relatively unique from the set of all possible digital images of that size and depth. As the overlap decreases between the views, the scene being captured is varying more and more, between the views, as parallax and occlusions begin to occur. The assumption that this is a small set of possible images since they are viewing the same scene is still valid; but the assertion that the views are still observing the same scene begins to fall apart. The PMFs of each view in the overlap as the cameras are rotated or separated by significant amounts become more independent and thus more separable and their possible mutual information decreases. It will be shown as a major characteristic of the evaluation of this algorithm and its robustness, to understand what factors and scenarios are crippling or limiting in terms of asserting dependence between the distinct views' PMFs. Also, As $\mathfrak{m}$ and $\mathfrak{n}$ decrease there is less opportunity for variation or consistency in the image, depending upon the value of b (the bit-depth). If 2^b is significant (*i.e.*not much, much less) compared to $\mathfrak{m} \cdot \mathfrak{n}$, then each instance of an image from that theoretical set can have significant variation. This could even allow for significant variation between views are capturing the same exact scene or sections of the views are capturing the same exact scene (spatio-temporally). This is a crippling limit of these assumptions and a source of our empirically understood minimum-overlap requirement for the algorithm.

2.3 Mutual Information

Excellent texts to reference for this discussion, pertaining to its development or when in need of a deeper understanding of tnformation theory, are [22], [23], [24], and [25]. The text by Cover and Thomas [25] is an excellent reference for those with a prior understanding of information theory but are in need of a refresher, while the text by MacKay [23] is better suited for those unfamiliar with information theory or the application of probability theory. A lot of the following discussion is best referenced by [25].

In a general case, mutual information can be understood as the measurement of the average

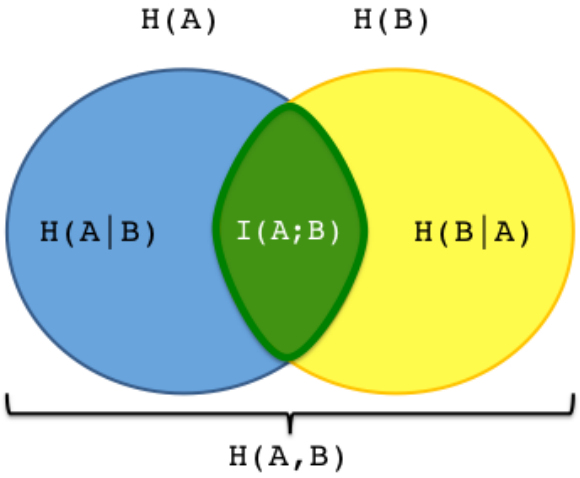


Figure 2.3: Venn diagram of key measures in Information Theory

shared determinism between two random variables; that is, how much does the probability of occurrence of one random variable portray about the probability of occurrence of another. In the case of digital images we are looking at discrete random variables, with their characterizing PMFs (probability of occurrence), and thus discrete entropy measures. Entropy, $H(A)$, is the measure of average randomness in a random variable and is used in the derivation of the mutual information formula. The following information theory terms to be described can be visualized in Figure 2.3. Calculating entropy for the random variable A is shown in Eq. 2.3. Calculating mutual information in digital images makes it a discrete measure based on discrete random variables and their distributions. Whenever you have two random variables, you can generate a measure of their mutual information, conceptually, by measuring how their marginal entropies relate to their conditional entropy. Again, by referencing Figure 2.3, it is quite clear that there are several ways to determine the mutual information $I(A;B)$. To choose the most appropriate method for calculating mutual information, a view of mutual information applied to image registration can make this derivation much clearer, albeit this is conceptually going in reverse.

$$H(A) = \sum_i p_A(a_i) \log_n \left(p_A(a_i)\right) \tag{2.3}$$

In beginning with two digital images, again with an example system image with 8-bpp and a single channel, the question of registration is: what pixels in image A are found in image B? A refinement of this question, thinking more in line with this research, is: what portions or regions, if any, from these images, have captured the same area of the scene? In general, as described in Section 1.2, the four stages of image registration typically deal with pixel correspondences. Once the correspondences are found, the corresponding pixel locations from one image are mapped to their equivalents in the other image, and that mapping is described as an invertible homography. The WFMI algorithm ultimately uses a discrete mutual information measure to determine the correspondences required to register the two images, but it does not look directly for pixel-to-pixel correspondence.

An 8-bpp $\mathfrak{m} \times \mathfrak{n}$ image is a bounded set of $\mathfrak{m} \cdot \mathfrak{n}$ intensity values distributed randomly in the interval of [0,255]. Corresponding this set to another image, *i.e.* another data set, requires some mathematical measure that is based on mathematical attributes of the data sets themselves, *i.e.* their PMFs. Given two digital images, their two intensity histograms can be found, then normalized to generate their image PMFs (also called the marginal PMFs), and then their marginal entropies can be determined through Eq. 2.3.

The measure of entropy is presenting the minimum number of bits required to encode or losslessly transmit the $\mathfrak{m} \cdot \mathfrak{n} \cdot \mathfrak{p}$ random pixels of an image. If they are completely random, the entropy is comparably large, but if the data is completely deterministic the entropy approaches 0. The mathematical determination of this randomness measure is essentially characterizing the amount (or lack thereof) of redundancy in the data. This gives a metric of encoding efficiency but not a means of encoding. Therefore it is best to understand entropy as a bounding metric, since it presents the minimum amount of bits for encoding in an ideal situation, which is often not achievable, or in the case of non-integer results, it is not physically possible (as we cannot realistically transmit 3.2 bits, but that is a potentially valid entropy). By equations 2.3 and 2.4, one can take the distribution of a data set and determine its entropy. Classically, n is taken to be 2, so as to create a value $H(A)$ measured as the total number of unique bits required to losslessly encode the data. Equation 2.4 is

the equation for generating the image PDF.

$$p_A(\mathbf{a}) = \mathbf{h_A}(\mathbf{a}) \cdot \left(\sum_{\mathbf{i}} \mathbf{h_A}(\mathbf{a_i}) \right)^{-\mathbf{1}} = \frac{\mathbf{h_A}(\mathbf{a})}{\mathfrak{m} \cdot \mathfrak{n}} \tag{2.4}$$

Now stepping back to the digital image as a data set, it is clear that a measure of entropy can be determined for each image. Then joint and conditional probability densities between the images can be determined from joint and marginal histograms. For the bivariate histogram, the (n_i, p_j)-bin's value means: there are this many pixels where image A has intensity relating to index n_i and image B has intensity relating to index p_j at the same spatial location (given images of the same size). In the case of generating the bivariate histogram between an image and itself, the result is an $\mathfrak{m} \times \mathfrak{n}$ "identity" matrix since an image will only have a value relating to index n_i where itself also has the value relating to index p_j when $i \equiv j$. For example, given $h_{AB}(0, 2) = 4$, then what is known is that somewhere in image A there are at least 4 pixels with intensity values in the range of bin 0, and there are at least 4 pixels in image B with intensity values in the range of bin 2, and that only 4 of those pixels in each image are in the exact same spatial locations within the images. That is all that is known and can be extracted from only the bivariate histogram. It too encodes spatial data that cannot be uniquely decoded.

This is a major factor in understanding the application of mutual information, because as is shown in Equation 2.5, the mutual information is developed from the joint and marginal histograms.

$$I(A, B) = \sum_j \sum_i p_{AB}(a_i, b_j) \log_2 \left(\frac{p_{AB}(a_i, b_j)}{p_A(a_i) p_B(b_j)} \right) \tag{2.5}$$

What is immediately apparent is that if the product of the marginal distributions is equal to the joint distribution, then the mutual information is zero. When a joint distribution is equivalent to the product of the marginal distributions the two random variables are said to be independent. So when the random variables are independent, then they will have no mutual information. And again, mutual information is, at its heart, a measure of randomness, i.e. an entropy measure. Joint

entropy depicts how random two images are, jointly, by denoting the number of bits required to encode them. Mutual information denotes how much the random information in A can convey knowledge about the random information in B. Thus in the further development of this algorithm the normalized mutual information value presented in Equation 2.6 will be used, as developed by the work in [26] which characterized it as a symmetric and normalized measure.

$$I(A,B) = \left(\frac{2}{H(A)+H(B)}\right) \cdot \sum_j \sum_i p_{AB}(a_i, b_j) \log_2 \left(\frac{p_{AB}(a_i, b_j)}{p_A(a_i) p_B(b_j)}\right) \tag{2.6}$$

What this provides is a mutual information measure that is dependent only upon the two images, A and B, and their total number of applicable pixels.

Now there is a single value result for any comparison of distributions, with high values meaning similar distributions and values approaching zero (as it is a non-negative measure) meaning dissimilar distributions. Since images or regions of images can be seen as random variables characterized by their distributions, then they too can be compared in this manner. Ultimately the question here will be how to generate the distributions, meaning, what data is used from the images to develop the histograms, and why it is done for these surveillance situations? Note that this is a distribution measure, not an element-by-element measure.

2.4 Camera Geometry

The previous Sections have presented the mathematics and concepts required for building the mutual information metric and applying it to digital images. The final major concept to understand in the theoretical development and motivation is how multiple views of a single scene can be related mathematically.

As the discussion has been already laid out, there are three distinct coordinate systems present in the mathematics discussed here. There are the scene coordinates, the camera (view) coordinates (the digital video), and the frame (image) coordinates. The camera view projects the real-world

scene onto the 2-D spatial domain of the camera plane while preserving the time dimension (albeit sampled and quantized in all of these dimension), and the frame is an instance of the time dimension of the camera. Video compression concerns will be ignored in this development though they play a role in the success of the algorithm, but simplistically they can be thought of as more quantization and interpolation. A diagram of the scene, views, and objects is presented in Figure 2.4.

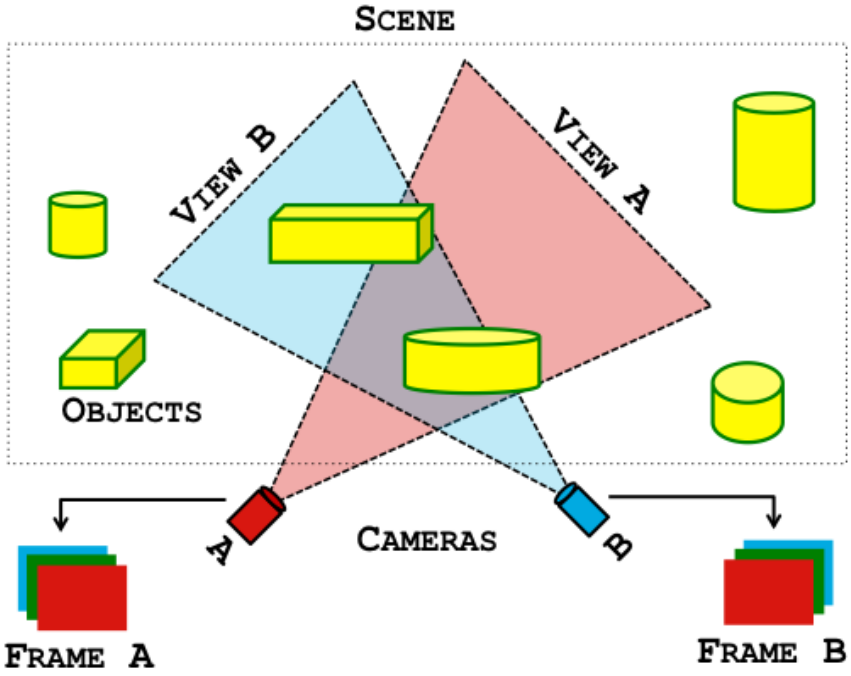


Figure 2.4: Terminology and structure of our application.

Understanding camera geometry is most easily understood by modeling the digital camera as a spatial projection system with a pinhole/point aperture [16]. Extending all of the following to realistic apertures would mostly just modify image spatial resolution and essentially creates a blurring effect on the results. Including this would only complicate the discussion as ray geometry would have to be abandoned for wavefront physics. So, again, the camera is thought of as a system that takes the light reflected and refracted off of the objects in the scene, allows it to pass through the point aperture (modeled as rays), and captures the light rays at the sample locations on the CCD array (which again has the Bayer filter in the case of color imagery). This model for a single

camera is shown in Figure 2.5.

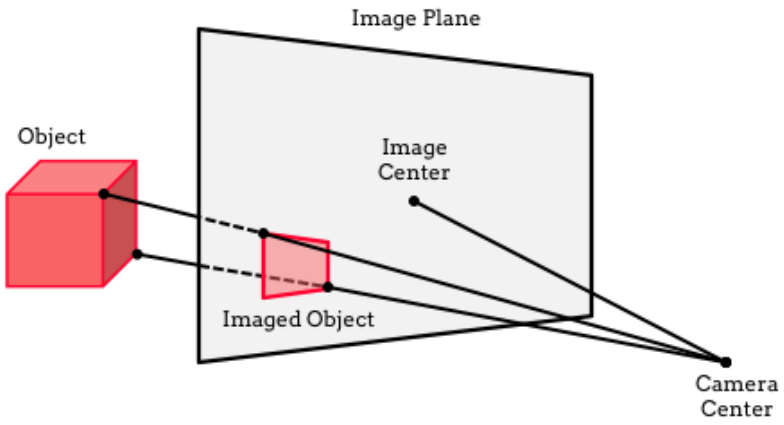


Figure 2.5: Projective Camera System Model

There are the 3-D objects in the scene that become projected through the camera aperture (not shown for simplification) onto the camera's image plane as a 2-D imaged object. As the camera center moves in relation to the object in the scene, the projection of the object will change. Looking at Figure 2.5, our first constraint/assumption is that the cameras (the views) will remain stationary but these other cameras viewing the same scene will have distinct projections of the objects in the scene. In this diagram the object is highly symmetric so there will be many views that seem identical but real-world objects are not typically this symmetric; although they do have typically rigid shapes with relatively smooth textures, *i.e.* large areas of low entropy. Clearly for multiple views, the closer the camera centers are, the more similar the views will be. If the scene has many objects and a lot of 3-dimensional spatial variation, then it will require less and less distance between camera centers for the views to change significantly. Yet modeling a general scene as having a significant but not extensive amount of 3-dimensional variation can fall back to the discussion on regional PMFs, where the overlapping regions in the views will be similar enough in intensity or projected scene content that their PMFs should have a higher mutual information than any other combination of regions.

In Figure 2.6 Camera A's image suffers from occlusion where the red block is covering a corner

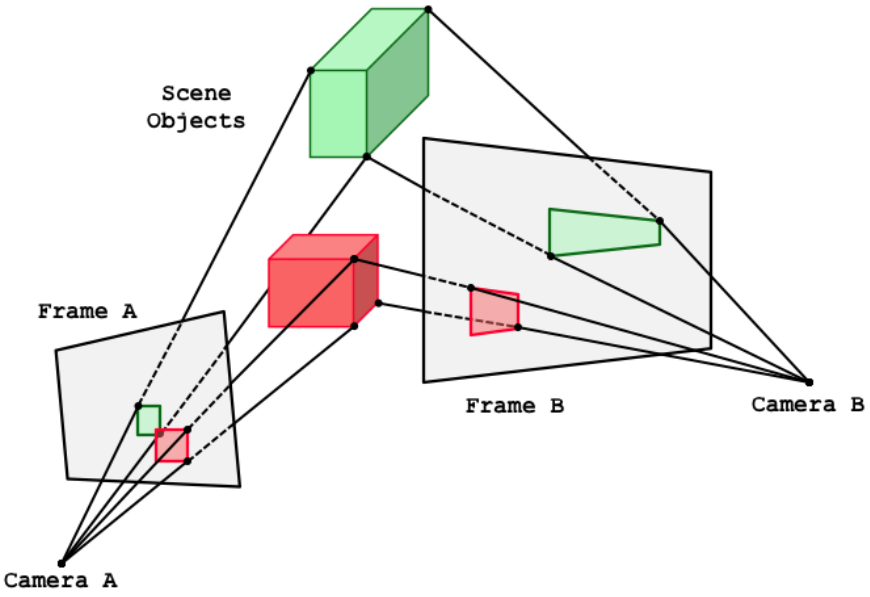


Figure 2.6: Multiple Projective Cameras Model

of the green block, as also indicated by the paths of the rays. There is also the effect of parallax, wherein with each view the distance between projected objects in the images varies based on the camera's distance to the objects in the scene. The red and green blocks appear very close together spatially in Camera A's view but are spaced far apart in Camera B's. Occlusion and parallax are not just characteristics of the views, they are characteristic of the entire scenario: the scene, its objects, and the views.

So ultimately what is applicable to this algorithm that needs to be understood about camera geometry? Projection is the main thrust of this discussion, even though multi-view geometry is an extremely rich and interesting topic, because the goal of this algorithm is to generate convincing panoramas of realistic scenes, automatically. Views like those modeled in Figure 2.6 are certainly relatable through epipolar constraints and use of a Fundamental matrix [16], but they would not produce a 2-D panorama. Clearly there must be some initial restriction, in the basic theory, where camera centers are not at angles of rotation much beyond 45^{o} to one another with respect to their views. Realistically, *i.e.* in practice, this is a general and typical scenario as cameras will be placed along typically rectangular structures such as buildings and fence lines or in hallways. Yet even

in these scenarios, there will be significant occurrences of occlusion, and differences in occlusion between views stemming from parallax discrepancies. With significant occlusion disparity, any point-to-point correspondence algorithm that considered occluded and non-occluded features would have added ambiguity in attempting correspondence in an already unknown scene. The more complex the scene is, such as irregular shaped objects and large variations in 3-dimensional arrangements, there will be many points existing in one view that do not exist in the other but could seem to, mathematically. This is the foundational motivation for designing the WFMI algorithm as a region-based correspondence rather than a point-based correspondence algorithm. Point-to-point correspondence can certainly allow for more realistic deformations, but it can also allow for extremely unrealistic deformations. Given that that corner of the green box was found in one view and not the other, it could be searched for in the second view and accidentally corresponded to some other box's corner in the scene. Using that relationship to develop the homography or transformation between the views [16] can produce extreme inaccuracy.

It is important to understand that occlusion and parallax are extremely difficult to model, but occur extremely often in realistic scenarios. With no known relationship between camera locations, the WFMI algorithm took this knowledge of multi-view geometry and its constraints to assume that even though there will not be a full set of points in an overlap region that can be corresponded, the structure and arrangement of objects will be there, and that's what should be searched for. This is why it is a region based algorithm, as it compares scenes based on the structure and arrangement of the objects, not the exact location of all the points on the objects. That is also why it is so successful despite only searching for an affine relationship between views more accurately relatable by a projective homography or a Fundamental matrix.

CHAPTER 3: ALGORITHM AND IMPLEMENTATION

With a robust understanding of the foundational mathematics and concepts, the discussion of the algorithm focuses on understanding the practicalities that molded the application of the theory. Limitations and the success of the results will be discussed in Chapter 4 with the results themselves. Each section of this chapter will break down the major components of the algorithm, focusing on the implementation details while drawing from the foundational mathematics discussed in the previous chapter. They will also present a "big picture" overview of the algorithm, discussing how the components from each section come together to form the fully functional algorithm. This allows a first-time reader to work from the nuts-and-bolts up to the big picture, while also providing a researcher the opportunity to reference this chapter very easily for review of the algorithm and its applied models. This chapter will also discuss the process of the actual implementation of the algorithm in MATLAB® and OpenCV alongside the details of the implementation.

3.1 Algorithm Components

The basic algorithm contains 5 major steps as shown in the flowchart in Figure 3.1 below. This chart shows the processing order of each step, but purposefully has labeled them by their goal and not their specific implementation details. The exact processes of the algorithm will be explained in the following subsections, but, as is the intended purpose of the flowchart, it should be kept in mind that the specific operations were chosen to serve the goals. The overall algorithm itself could be implemented differently as long as the main goals in each step are met. This will be discussed further in Chapter 5 as a possibility for future work in modifying some of the chosen means of implementing these components. The story of the development of the implementation is presented alongside the purpose and method of implementation, as they are interrelated.

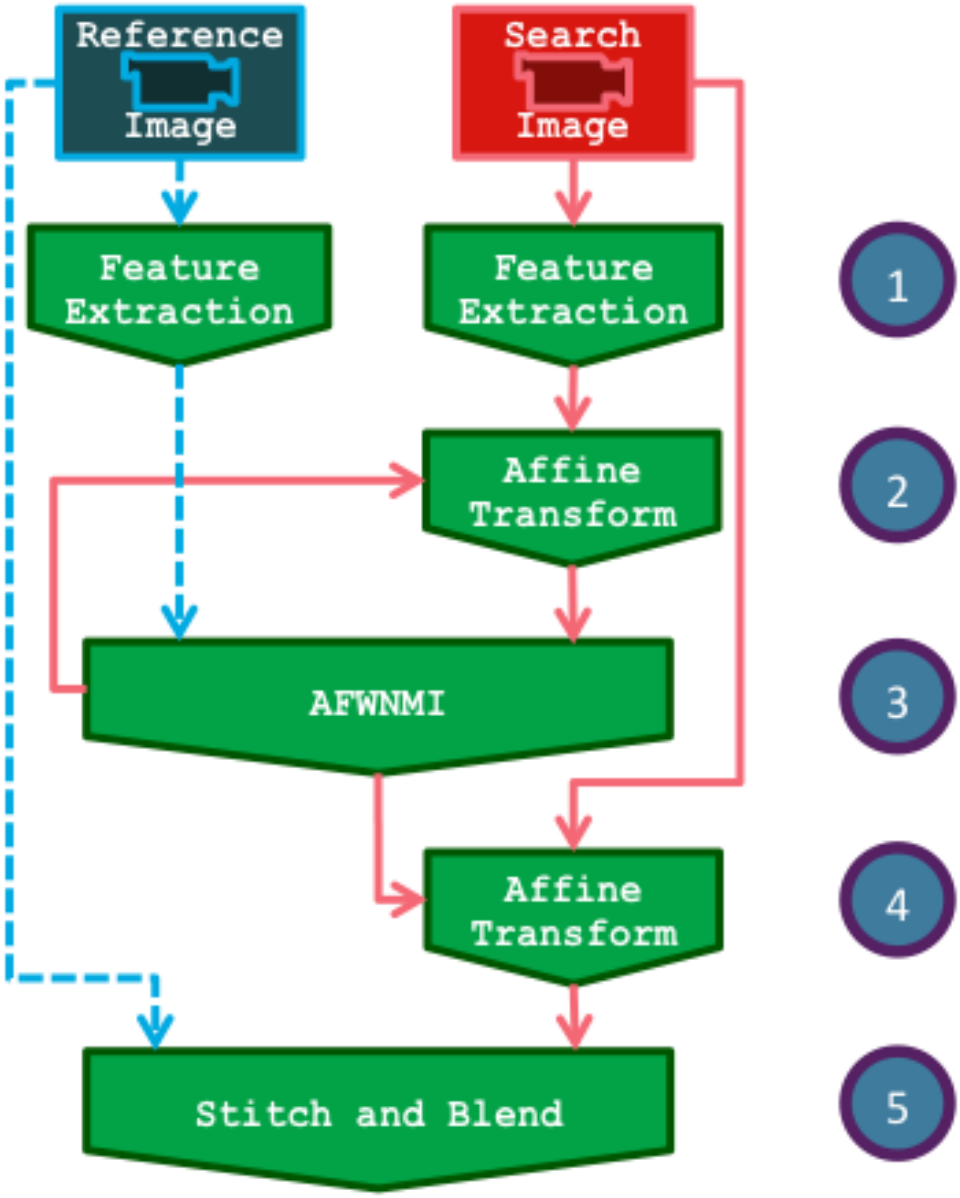


Figure 3.1: Algorithm Flowchart

3.2 Feature Extraction

For the WFMI algorithm it was reasoned that the initial stage of feature extraction should be simple enough to allow the eventual possibility of real-time operation, but must still provide robust enough features to secure success in a variety of complex and unknown scenarios. Looking to the work in [27] the choice of color gradient features was found appropriate to provide a robust set of non-rigid shape details that, even under scaling and quantization of the image(s), preserve a significant amount of the original information about the scene structure and content. After generating the RGB-gradient map, a quantization step was implemented to achieve a computationally efficient algorithm, while also binning features based on the type of structure they represented (both empirically and conceptually).

Given an $\mathfrak{m} \times \mathfrak{n}$ 8-bpp pair of 3-channel color images, each image (developed with 480×640 RGB images) is first processed to extract a feature map. The algorithm from [28] is a vector-space gradient operation calculated by using the vectorized color pixel values ($\mathfrak{p} \times 1$ vectors) as locations in the current color space. Taking horizontal and vertical gradients along each channel in the intensity domain and then using those resultant gradient images as inputs to the calculation of the maximum eigenvalue of the space-to-color vector field matrix ($D^T D$ in [28]'s notation, Eq. 3.4) provides an essentially infinite, floating-point range of intensity values in the resultant color gradient map. This will be a single-channel map of size $\mathfrak{m} \times \mathfrak{n}$, matching the original image channel dimensions. In a practical implementation, there is an initial quantization at this point to limit this theoretically infinite floating-point map to the bit-depth of data-type. In the MATLAB® and OpenCV implementations this was set to 64-bits, since this data holds the maximum amount of information that can be generated or stored by these features. Any quantization operation beyond this will superfluously remove information, but will enhance computational efficiency in the PMF calculation which has a range of 2^b (with b being the bit-depth) maximum bins; so the key is to then discern what amount of quantization will maintain enough information to generate an appropriate and accurate mutual information map that is applicable for further processing in the registration search.

The subsequent step of the algorithm is to perform an affine transform search with an exhaustive translation search. This will be explained in Section 3.3, but in order to perform a computationally efficient search with a metric based on joint and marginal PMF calculations for each translation in the exhaustive search, there is a need to quantize the color gradient maps into the "edge" maps of b_e-bpp. The optimal implementation for this secondary quantization was found with maps on the order of 1- to 4-bpp thus providing on the order of three to thirty distinct "edges". Since the PMF calculations will be based on full range histograms, then the joint PMF size will be $2^{b_e} \times 2^{b_e}$, where b_e is the bit-depth of the gradient to "edge" quantization. Note that the use of "edge", in scare-quotes, is to maintain that this is not a binary map. There is no thresholding operation, as the gradient is simply quantized, no binarized.

The stronger the quantization, *i.e.* the fewer the number of "edges", the less information available in the image from the color-space gradient map, yet this is inversely proportional to the computational efficiency. Also, discussed in more detail later on, in the use of a hierarchical search there is an added concern as to the effects of the subsampling on the image information. These are all significant implementation concerns, and it was found that the fastest and most accurate implementation was achieved with $b_e = 2$, thus having 4 significant "edges" while zero values were ignored in the PMF calculations. To determine the quantization, a simplification of the method in [27] was used.

The top 20% of the pixels were empirically determined as the strong object "edges", with the bottom 10% being determined as noise or superfluous detail "edges". The remaining 70% of the pixels were evenly split into $b_e - 1$ groups of so-called significant detail "edges". As the gradient histogram was found to typically follow a low-variance Poisson-like distribution with right skew. These percentage bounds were found to empirically average the distribution rise with the bottom 10%, the long tail with the top 20%, and the majority of the distribution with the middle 70%. This was an empirical development that is fast, efficient, and has been found to provide useful and conceptually valid "edges" in the realistic test images. A visualization of this quantization is shown in Figure 3.2.

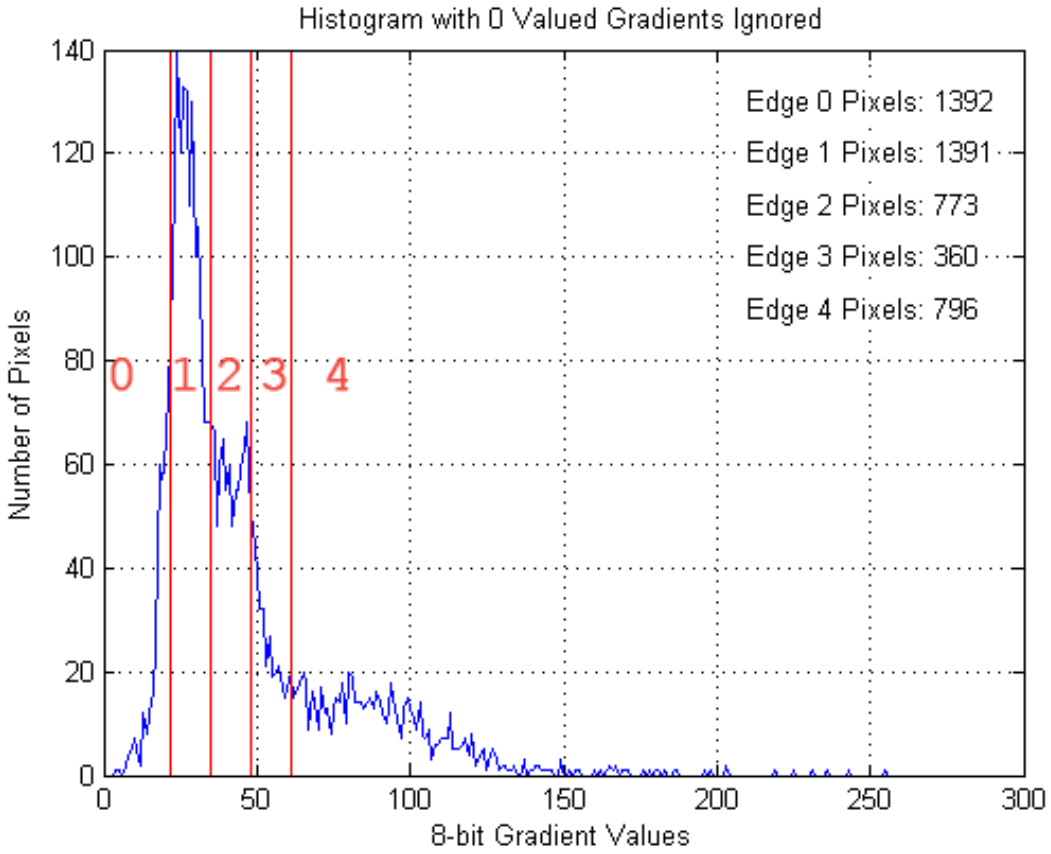


Figure 3.2: Example Gradient Map Histogram with Quantization Boundaries

$$p = \left(\frac{\partial R}{\partial x}\right)^2 + \left(\frac{\partial G}{\partial x}\right)^2 + \left(\frac{\partial B}{\partial x}\right)^2 \tag{3.1}$$

$$q = \left(\frac{\partial R}{\partial y}\right)^2 + \left(\frac{\partial G}{\partial y}\right)^2 + \left(\frac{\partial B}{\partial y}\right)^2 \tag{3.2}$$

$$t = \frac{\partial R}{\partial x} \cdot \frac{\partial R}{\partial y} + \frac{\partial G}{\partial x} \cdot \frac{\partial G}{\partial y} + \frac{\partial B}{\partial x} \cdot \frac{\partial B}{\partial y} \tag{3.3}$$

In order to generate the gradient map, as characterized by the distribution in Figure 3.2, the maps of Equations 3.1-3.3 must be calculated. In the MATLAB® and OpenCV implementations, the Sobel operator was chosen to generate the partial derivatives in the x and y directions. These

maps, as mentioned, make up the components (x, y)-space to (R, G, B)-space vector field matrix as shown in Equation 3.4.

$$D^T D = \begin{bmatrix} p & t \\ t & q \end{bmatrix} \tag{3.4}$$

The gradient map is then calculated, at each pixel location, by taking that pixel location's $D^T D$ array and finding its maximum eigenvalue. The achievements of Lee and Cok [28] simplify this by producing a closed-form solution to that array's maximum eigenvalue. Therefore, through the use of Equation 3.5, there is no need for any Singular Value Decomposition or Eigenvalue determination, and through array operations the gradient map is directly calculated as $\tilde{\lambda}$.

$$\tilde{\lambda} = \sqrt{\frac{1}{2} \cdot \left(p + q + \sqrt{(p+q)^2 - 4 \cdot (p \cdot q - t^2)} \right)} \tag{3.5}$$

The $\tilde{\lambda}$ map/image is normalized to its peak value and quantized (if not explicitly done so) to a 64-bpp array. The notation of these equations follows [28] directly, to avoid confusion. Again, this is then built off of only two convolution operations, the vertical and horizontal Sobel kernel, and the map itself can be directly calculate without any Eigenvalue or SVD operations, thereby avoiding the calculation of an $\mathfrak{m} \times \mathfrak{n}$ matrix (pseudo-)inverse.

3.3 Affine Transform Search

The affine transformation is defined by 4 distinct operations: scale, skew, rotation, and translation. Skew, translation, and scale can each be defined separately for the rows and for the columns of the image. This allows for more complex deformations but from the understanding of projective geometry, it would be more accurate to model realistic deformations by different amounts of scaling or skew variation across the view, not just a constant transformation equally along all the rows versus another equally across all the columns. This approach was not investigated because the current implementation of the algorithm already performs an exhaustive translation search and

research into more complex transformations proved to unacceptably increase computation costs.

The most drastic assumption made by the WFMI algorithm is in the utilization of an affine-only search for image registration in views with relationships significantly outside affine constraints. This was an assumption made mostly for computational efficiency and simplicity, it is not an assumption based on a strong mathematical or conceptual understanding such as the assumptions discussed in Chapter 2. That being said, it is an efficient and robust means of generating estimates for accurate registration between overlapping views of complex scenes. In the actual implementation the affine transform was not usually implemented but a similarity transform (rotation, scale, and translation only) was used instead. It was found that in most scenarios, skew had little to no effect on providing a more convincing view and so that dimension of the search space was ignored. That is in practice only; in the design of the algorithm it is part of the full implementation. It was also found that between rotation and scaling, rotation proved the more important factor in creating a convincing view. The reasons behind this will be discussed with the results presented in Chapter 4, as the rest of the algorithm must be explored first.

The affine (or similarity) transform search has been implemented in practice as a limited search in rotation, scale, and skew, with an exhaustive translation search. While it is maintained that this is a fully automatic algorithm, there is the practical consideration that between two overlapping views of a realistic scene there will not be views rotated by 90º between them or scaled by 100%. With that understanding, then, it was found through testing that realistic scenes with average overlap (usually 20-40%) were in the range of -15^o to 15^o of rotation and a scale factor of 0.8 to 1.2 was a generous range. While skew was usually ignored in the actual implementation, it can be an overcompensating factor and so while a range similar to the scale factor seems reasonable, it was usually restricted to about half that range, to maintain more reasonable results. The main problem with skew is that when deforming an entire image, it did not prove to be a reliable estimate for the effects of parallax and projective distortion in views of realistic scenes. Again, more of the algorithm needs to be discussed first, so this will be elaborated on in the following chapter, but skew tended to promote mis-registration of low entropy regions as it allows the view to distort in

an unrealistic manner.

The search itself is done as two stages. First, one image is chosen as the reference image; the affine homography to be found will be a transformation for the second image (the search image) onto this first image (the reference image). The homography should be invertible and so the reference image could be transformed onto the search image using the inverse of the transform matrix, but this is more of a mathematical consideration than a practical one [16]. In the case of a full affine search, the skew, scale, and rotation ranges are pre-defined (as was mentioned, this is a practical limitation) and the search space is the three-dimensional space covered by these ranges. At this point translation is being ignored. Once the reference image (following the left path of Figure 3.1) is chosen and the affine (without translation) search space is defined, the second image (following the right path of Figure 3.1) undergoes a transformation in this space. For this implementation there was no enhancement made to the navigation of the search space and so the skew-scale-rotation-space (affine search space) is searched linearly, which may not be the most accurate method. Once the second image has undergone the transformation it is now a temporary new image and it is passed to the correspondence stage (Section 3.4) along with the unmodified reference image for comparison. Note, this is done at the top level of the Gaussian pyramid created for the hierarchical search, and this is done on the 64-bit color gradient image maps.

The flowchart of Figure 3.1 is conceptually accurate but in practice, especially during the transition from MATLAB® to OpenCV, it was found that the Gaussian pyramid reduction adds superfluous quantization to the "edge" images. This was found in the OpenCV implementation because C++ is a hard-typed language and MATLAB® defaults to 64-bit data; so in the prototyping stage it was not initially recognized that the b_e-bit "edge" images were actually being stored as 64-bit images. What this requires then is to first build the Gaussian pyramid with the 64-bit gradient maps, choose the reference and search images, transform the search gradient map, and once it is transformed it can quantized into the b_e-bit "edge" map. After choosing the reference image, it can be quantized as well. The correspondence stage (Section 3.4) is thus working on the reference image's "edge" map and the transformed search image's "edge" map, which are from the top of the

Gaussian pyramid for the hierarchical search. In both implementations these images are padded with zero values (since zero is ignored as data) to be the same size.

The next section will discuss the implementation of translation, but a final remark is to note that in the overview of the algorithm it is important to keep in mind that the correspondence is occurring in the skew-scale-rotation-space. So for every combination of skew, scale, and rotation (every point in that discrete 3-D space) a peak from the filtered and weighted mutual information map will be found and will be compared to the rest of the space. In implementation, to save memory, each time the temporary image is created by the transformation of the search image, it is replaced when moving to the next transformation in the affine search space. In Figure 3.1 there is a second transformation after the WFMI stage (the correspondence search), and this second transformation is defining the final transformation for the search image based on the results from the correspondence search. Again, since each transformation was thrown out during the search, once the optimal peak is found from the affine search, the corresponding affine transform must be reapplied to the search image, while the reference image still remains unmodified.

3.4 Weighted and Filtered Mutual Information

This stage of the algorithm is part of the affine search stage, so it should be kept in mind that the following calculations are re-occurring at each point in the search through the skew-scale-rotation-space. This must be the most efficient stage in the algorithm because it occurs so many times, but it is also the core of the algorithm that is the limitation for the accuracy. This section is the most important in understanding the trade-offs between accuracy and efficiency.

Up to this point the discussion of mutual information has described it as a measure based on random variables. Those random variables were then attributed to the digital color images. But in the actual implementation, it is performed on the maps of the "edge" values (in the range of 2^{b_e}) from each view. But it is not just the mutual information between the two images that is being calculated, it is the mutual information between every possible rectangular overlap between the two images (in an exhaustive translation search over the "edge" maps). This makes the mutual

information a mapped measurement, meaning that the value of the peak of the mutual information is mapped to a particular pair of (row, column) translation parameters. This is not a direct mapping since the mutual information measure is over the $(2 \cdot \mathfrak{m} - 1) \times (2 \cdot \mathfrak{n} - 1)$ space. Ultimately a shift of $(\mathfrak{m}, \mathfrak{n})$ is made to determine the actual translation parameters. To simplify the notation, the mutual information map is said to be of size $M \times N = (2\mathfrak{m} - 1) \times (2\mathfrak{n} - 1)$.

So, the search image has been transformed and the "edge" map has been extracted, along with the "edge" map for the reference image, and these are passed into the correspondence stage (labeled WFMI in Figure 3.1). The most efficient implementation was achieved by creating arrays of indices that correspond to two corners of the overlap region for each "edge" map. Since all the overlaps will be rectangular, knowledge of the top left corner point and bottom right corner point can determine the entire set of pixels in that rectangular region. The initial implementation tried comparing regions based on circularly shifting through the maps, but this meant that at every calculation, even when only 1 pixel was being compared, the entire array had to be passed along in memory. And actually to perform the circular shifting accurately, the arrays had to be padded to $2\mathfrak{m} \times 2\mathfrak{n}$ with null values. By inspecting the operation and utilizing the sizes of the arrays, the full set of overlaps can be found by their two corner pixel locations (as mentioned), and all these indices can be stored in a set of arrays. Then in the actual exhaustive translation search, only those pixels in the bounds of the indices are extracted and passed through the calculations. This is far more efficient, albeit a more conceptually complex means of implementation. It is much easier to understand the following calculations while thinking of the arrays as layers overlapping in space as they shift pixel-by-pixel creating every possible rectangular overlap.

Now having the pixels that are in the potential overlap, the mutual information between the two "edge" maps in this overlap must be calculated, and this will be the metric for judging correspondence. This requires calculating the bivariate and marginal histograms with the two maps. This was also an initially slow operation, as there was no optimized implementation of a bivariate histogram in MATLAB® or OpenCV. Obviously the bivariate histogram cannot be generated from the marginal histograms unless the two regions are independent, but if they're independent their

mutual information will be null, so this would be a completely counter-productive assumption. However, the marginal histograms can be calculated from the bivariate histogram, regardless of the independence of the two maps' data, if a full range bivariate histogram is being implemented. This must be inspected carefully in implementation, because while the full set of "edge" values are relevant to the bivariate and marginal histograms' calculations, 0 values are being ignored in the algorithm as "not data". If the implementation of the bivariate histogram is only $2^{b_e} \times 2^{b_e}$ then it has not counted any zero values, thus it also has not counted any pixels from the reference or search map that line up with zeros in the other map. Generating the marginal, or univariate, histograms from this bivariate histogram would be a glaring mistake that ignores a possibly significant amount of data. Conceptually it actually would push the algorithm towards registration in extremely low entropy or non-existent data regions. Again, the zero values (null values) signify either the presence of a noise or extremely fine detail "edge", or a region outside the bounds of the image after the transformation, because the images are padded by the null values to be the same size, which makes the complex indexing optimization simpler to implement. The actual implementation of the bivariate histogram in the algorithm used binary, element-wise multiplication and then a summation of a modified version of the "edge" maps. Essentially for a range of 2^{b_e} there are 2^{b_e} maps built where each of these new maps is a binary image. The first image will have a value of true (1) where it is equal to 1 (the first edge value) and all other pixels in the map will have a value of zero. This follows all the way up to the 2^{b_e} "edge" value. Having these 2^{b_e} maps for both images' "edge" maps allows direct element-to-element multiplication, and then the resultant multiplied map can be summed and that total value (the addition of all the ones) will signify the amount of values that exist in the corresponding locations, thus filling in the bivariate histogram. Take the first "edge" value map for example: in both the reference and search map there will only be a 1 where that first value exists, everything else is zero. Since $1 \cdot 1 = 1$ and $0 \cdot 1 = 0$, the resultant multiplied map will only have a 1 where both maps had a 1 in the same relative location. The phrase *relative location* is used because these are actually the extracted regions of potential overlap (this is all still in the exhaustive translation search), so what is being found is: for this given overlap scenario, how many

pixels correspond between the maps in these overlap regions? So to build the $2^{b_e} \times 2^{b_e}$ bivariate histogram, there will be $2^{b_e} \cdot 2^{b_e}$ multiplied maps and their summations. Given that this is being performed at the top of the Gaussian pyramid and that these are now sets of binary data (storable as 1-bpp), this can be an extremely efficient calculation, and in 2^{b_e} of those operations the marginal histograms can be found without having to append an extra iteration.

Now the bivariate and marginal histograms have been found and they can be normalized to become the joint and marginal PMFs for that overlap region. What also can be calculated from the bivariate histogram is the number of "edge" pixels that were found in the overlap region. Using the bivariate histogram takes an accurate look at only the pixels that do not correspond to null regions in the overlaps. This is the opposite of the discussion above, because this count will be used as the weighting for the accuracy/significance of the mutual information map. This is illustrated by Equation 3.6.

$$w(A, B) = \sum_{j} \sum_{i} h_{AB}(a_i, b_j) \tag{3.6}$$

The use of the bivariate histogram limits the effect of regions of overlap containing lots of null data. Null data will limit the number of "edge" pixels being counted, since the bivariate measure does not count when an "edge" in one map lines up with a null in the other map, and so it is statistically possible to produce false maximums for the mutual information. So the fewer the number of pixels used in the calculation, the more likely they are to correspond, and thus the less likely they are to be true correspondences. Essentially the weighting map will be an $M \times N$ deformed pyramid shape. The center pixel in the $M \times N$ weighting map (and mutual information map) corresponds to both images overlapping each other completely. This is the point with the most features in its calculation, so it is the most statistically unlikely to correspond, and thus must be compensated to compare accurately. This may seem counter-intuitive, but consider again that there are noise and intensity variations present across the views, as well as all of the discussion of the quantizations made throughout a digital image system (as presented in Chapter 2). Even if the same camera takes two images in succession without any motion, it is statistically unlikely that every single

pixel value will be identical, even though conceptually they should be. This is again why these "edges" are used for correspondence rather than just intensity values, because scene shapes should be maintained under realistic noise and variations. So this weighting is crucial to the success of the algorithm, otherwise it is statistically likely that all the small overlap regions (a few rows or columns overlapping) will produce peaks in the mutual information map.

$$I_w(A, B) = w(A, B) \cdot I(A, B) \tag{3.7}$$

Now with the PMFs defined and the weighting map, just as presented in Chapter 2, the normalized mutual information (Equation 2.6) can be calculated, and then it can be weighted by the pixel weighting map as presented in Equation 3.7 producing the Weighted Normalized Mutual Information (WNMI). Normally with the WNMI map, a peak is searched for, the pixel location relating to the peak is related through a shifting operation back to the translation parameters required to produce the overlap and the full affine transformation would be defined. However, as will be shown in the results in the next chapter, it was found that this did not take into the statistics of realistic views. For affine views the peak of the WNMI map would usually be accurate, but it was found that even for affine views, significant noise or low entropy could actually create statistically-accurate false peaks in the map. These will be discussed more in the next chapter. What was required though was to perform an operation that could extract not just the maximum peak in the WNMI map, but the maximum *and* sharpest peak. The rationale stemming from an understanding of what occurs when identical data is passed through a mutual information (or even a correlation) metric. To step back to a theoretical understanding, the autocorrelation (or correlating a signal with itself) of a set of data produces impulse-like signal, where the only shift that can cause a peak is when the image is aligned with itself. For this discussion, the mutual information map (WNMI map) would have a single peak at its center, when the entire image is overlapped with itself without any shift or transformation. This happens because every single feature value aligns and the PMFs are identical and so with any shift in the data, there will be a significant loss of alignment thus a drastic change in the joint PMF. So when correlating, or finding a mutual information map for, an image (or signal)

with itself, it will have a very strong peak when aligned and will drop off extremely rapidly for any shift. For a more complete mathematical understanding, this is identical to the theories behind "matched filter" design. So, even if the two views aren't identical, even in the overlap region, they should be more similar than anything else in the images; but more importantly they will create a peak when aligned and any small shifts in that alignment will produce significantly reduced values (comparatively). So even in a low entropy case when the center of the WNMI map will produce a statistical mountain/plateau, there will be a peak somewhere in the map that is very narrow and corresponds to the true optimal translation. Thus by filtering the WNMI map with some peak detector operation, that very thin but comparatively tall peak will be raised up while slowly varying regions, flat regions and mound-like peaks, will be lowered. Thus the implementation of the Laplacian kernel as given in Equation 3.8. This kernel looks like an upside-down traffic cone and is an excellent peak detector.

$$L(u,v) = \frac{4}{(\alpha+1)} \cdot \begin{bmatrix} \frac{\alpha}{4} & \frac{(1-\alpha)}{4} & \frac{\alpha}{4} \\ \frac{(1-\alpha)}{4} & -1 & \frac{(1-\alpha)}{4} \\ \frac{\alpha}{4} & \frac{(1-\alpha)}{4} & \frac{\alpha}{4} \end{bmatrix}, \quad \alpha \in [0,1) \tag{3.8}$$

As you can see from the kernel, the neighboring values are compared to the center value (the value being filtered). So if the region is flat, this zero-phase FIR filter will produce a zero value. But if the center value varies significantly from its neighbors then there will be a large value produced (positively or negatively). As the Laplacian filter can also be thought of as a second derivative filter, there will be both positive and negative values produced as the kernels rises along and then descends down a peak, thus is is necessary to actually take the absolute value of the filtered WNMI map, producing the Absolute value of the Filtered, Weighted, Normalized Mutual Information (AFWNMI) map. A peak in the AFWNMI map now corresponds to the maximum, most impulse-like, peak in the WNMI map, and is far more statistically likely to correspond to the optimal translation parameters.

The last step to mention is that the peak location from the AFWNMI map is not directly the

translation parameters needed for the affine transformation. A few algebraic steps are required to work back through the process and determine that actual translation that has occurred. The peak location is express by the values in Equation 3.9, where the (u, v) space denotes that the AFWNMI map space is not the same as the image space, which has been referred to as (x, y).

$$(\delta_u, \delta_v) = \max_{u,v} (\mathfrak{I}_{L\mathfrak{w}}) \tag{3.9}$$

With these values for the peak location, the shift for the map size must be accounted for and removed, as shown in Equation 3.10. This equation is shown with the values as point pairs but this can be implemented as vector operations in MATLAB® or performed separately in a non-array language.

$$(t_x, t_y) = (k, l) \cdot |(\delta_x, \delta_y) - (\mathfrak{m}, \mathfrak{n})| \tag{3.10}$$

The values k and l are used as the sign of the translation, because the de-shifting operation does not accurately produce the proper sign for the translation, so a second operation (as shown in Equation 3.11) is required.

$$(k, l) = \begin{Bmatrix} -1, & (\delta_x, \delta_y) \leq (\mathfrak{m}, \mathfrak{n}) \\ +1, & (\delta_x, \delta_y) > (\mathfrak{m}, \mathfrak{n}) \end{Bmatrix} \tag{3.11}$$

The goal is to set k and l to be either -1 or 1, corresponding to the sign of the translation in that direction. The assumption is that the transformed search image will be in the space of the reference image, and the image centers will be the centers of the coordinate system, for the purpose of the transformation (different from indexing coordinates). So since the AFNMI map center is at $(\mathfrak{m}, \mathfrak{n})$, then being less than that would be a shift left or up, negative, and otherwise it's a positive shift to the right or down. The actual implementation, as will be discussed in the next sections, resizes the images to padded versions where they can be thought of as layers, and so the values of k and l are calculated slightly differently to choose the appropriate padding amount and direction, but this

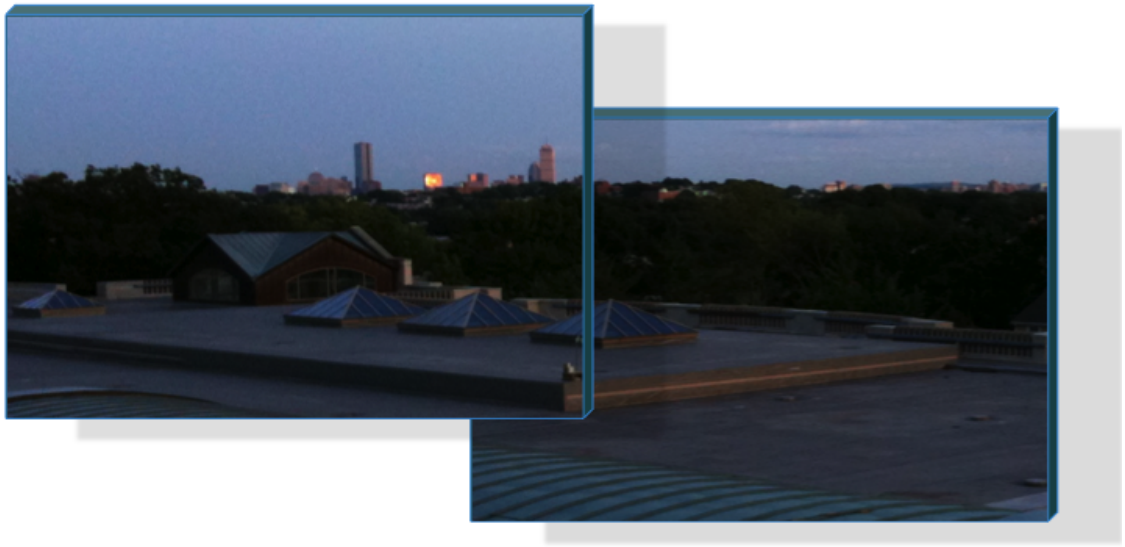


Figure 3.3: Depiction of Images as Layers in single Image Coordinate Space

displays the concept more succinctly and generally.

3.5 Stitching and Blending

At this point in the algorithm, following along in Figure 3.1, the optimal affine homography has been found for the search image to allow it to be transformed into the image space of the reference image. Now with these two color images, transformed appropriately, the problem being faced is how to blend these views now that they are overlapping. Conceptually (ignoring the 3 color channels) the images can be seen as two layers overlapping each other in the image space of the reference image, no longer in disparate image spaces. Thinking of these images as layers, there will be one image in front of the other, covering a portion of the second image, and that portion being covered is the overlap region, as depicted in Figure 3.3.

The simplest method would be to avoid the blending, cut the images at some halfway point in the overlap region, and place them side by side, essentially "flatten" the layers. The problem this presents is that if the homography that was found is not perfect, the stitch line will be clearly visible as objects in the views of the scene will not be spatially coherent across the seam. Common

errors will be duplicated objects (“ghosting”) or jagged breaks (“jumps”) in the combined view of the scene. If a more accurate homography could be found, perhaps through an iterative process or through the use of reapplying the algorithm to objects in motion in the scene, then the “ghosting” or “jumps” could be minimized. Yet, these errors will only be non-existent if the homography is perfectly accurate for the views, overcoming all parallax and occlusion disparity concerns. And then there is another possible source of error, beyond the spatial concerns, in the potential illumination variation and environmental artifacts (weather, *et cetera*).

Using views at unknown locations in realistic scenarios provides a huge range of variability in the artifacts, noise, and illumination source(s) for the views. For example, in an outdoor scene with cameras placed on a low rooftop, there could be a camera placed by/under a tree while the other camera is out in the open. When there is inclement weather or at certain times of the day (changing position of the sun), the views will have very different illuminations and artifacts. So, even if an accurate homography (or accurate estimate) is produced, there needs to be a method to automatically overcome these artifacts, otherwise the view will be very unconvincing. But given that these realistic scenes are well-beyond the affine constraints, and the transformation applied is only an affine transformation, there is nothing to support any assumption of a perfectly accurate homography in a realistic scenario. That means that before the algorithm even runs, it is known that at the outcome of the registration, there is no guarantee of pixel correspondence. Without pixel correspondence in the overlap region, it becomes an ill-posed problem to determine the illumination disparity between the views accurately.

An example to illustrate the scenario is to think of overlapping views of a scene with a red pickup truck in it. Once these views are registered and the search image is transformed into the reference image space, the overlap region would ideally contain the same view of the same red truck. Using the concept of layering the images in reference image space, the pixels on the truck should be in line in this third dimension (the layer dimension), and so with any weather or illumination artifacts/disparities between the views, the corresponding pixels on the truck are desired to be at the same color value. So, very easily a difference could be found between these pixels

and a color mean could be determined and the two views could be adjusted to this mean before they are blended, so that they appear to have the same illumination and weather artifacts that affected the illumination will have been overcome. But the WFMI algorithm produces an estimate of the overlap region because of the complications presented by the positions of the views and the structure of the scene. So, in thinking of these images in the layer-space, that means that the red truck won't line up pixel-to-pixel, but the disparate views of the red truck will now be in the same region. Trying to take color-mean differences would require knowing which pixels to use, but that is exactly what is not available. Understanding the structure and nature of the algorithm and the views pushed the development towards the implementation of the multi-resolution spline blending algorithm, which works based on a frequency content blending of any two views. The scene content, the red truck, is aligned through an estimate, so its frequency content should be relatively similar, or it is desired to be so. Thus by blending in the frequency domain, the misaligned spatial structure becomes a secondary concern, and is no longer a hindrance to creating a convincing view in terms of illumination and contrast variations.

When implementing the multi-resolution spline blending algorithm from [17] in the WFMI algorithm, first the the search image is transformed into the reference image space. To aid implementation the images are padded with zero values (the ignored/null values) so that if they were thought of as layers, then they would be the same size and the translation between the views is implemented directly by the zero-padding. See Figure 3.4 for a visual representation of the padding and a transparent overlapping view.

This makes computation simple because the images are now the same size and the location of the seam will be complimentary with respect to the columns and/or rows in each image. However, this does add in a significant amount of new data, even though all that data is zero. But again, the multi-resolution spline blending algorithm can be thought of as a decaying illumination mean adjustment, and now both images have had a potentially drastic mean shift because of all the zero values added to the image content. What is observed in the results is that the images are slightly darker, and there is superfluous blending from each image into the zero padded regions of the other

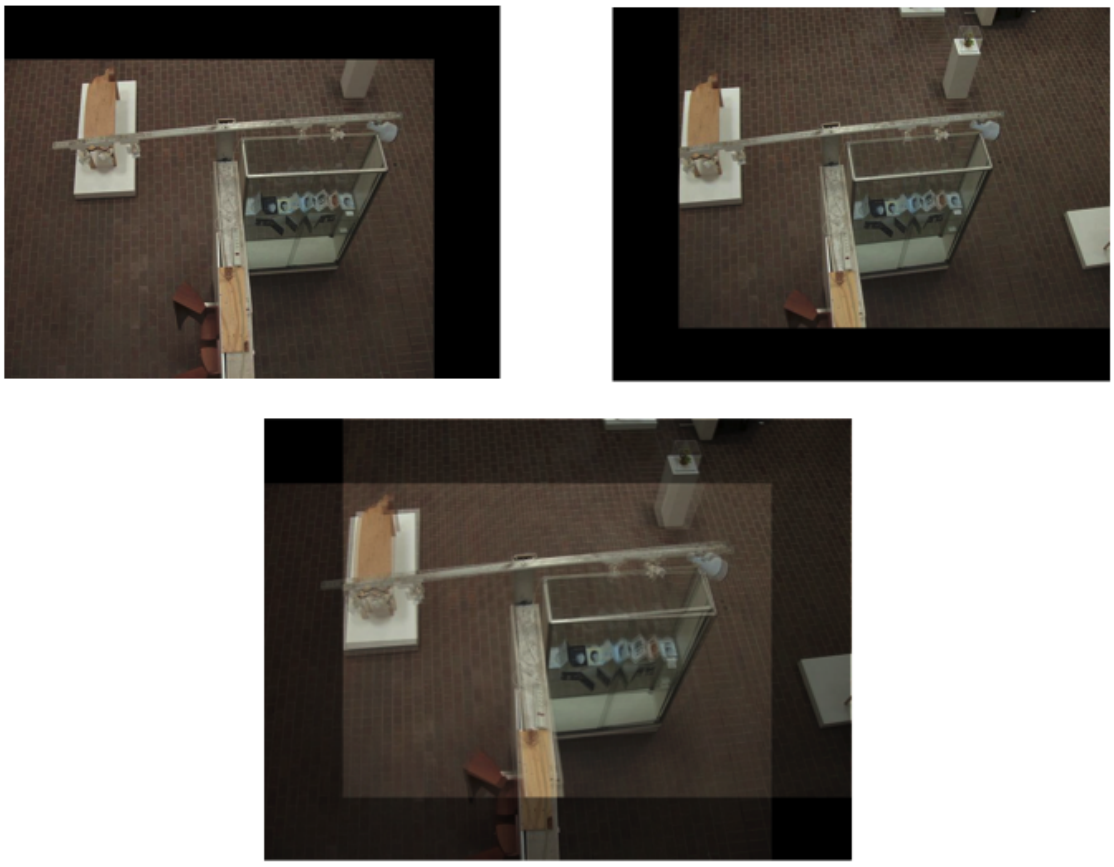


Figure 3.4: Rooftop Scene (a) Left View Padded, (b) Right View Padded, (c) Overlapping Padded and Transformed Views

image. These are implementation artifacts that do not cause significant detriment to the results and over-complicating the algorithm to avoid them was not a priority in this initial development. The registration process for multiple images would be to register two images, then continue through the other images adding them on to the image(s) that are already registered. Though consideration must be made if data is thrown out in any initial registration, as subsequent registrations may not be available because they might have corresponded to those lost regions.

The multi-resolution spline algorithm is based on a Laplacian pyramid reduction, generated by building a Gaussian pyramid, and then taking the difference between pairs of layers. The bottom layer of a Laplacian pyramid is the difference between the bottom two levels of the Gaussian pyramid. Since the Gaussian pyramid layers are subsampled, the smaller of the two layers will

need to be upsampled in order to perform a pixel-to-pixel difference between the layers. Once the Laplacian pyramid is defined for each image, a stitching seam is defined and the images are cut and combined along that seam, as shown in Figure 3.5.

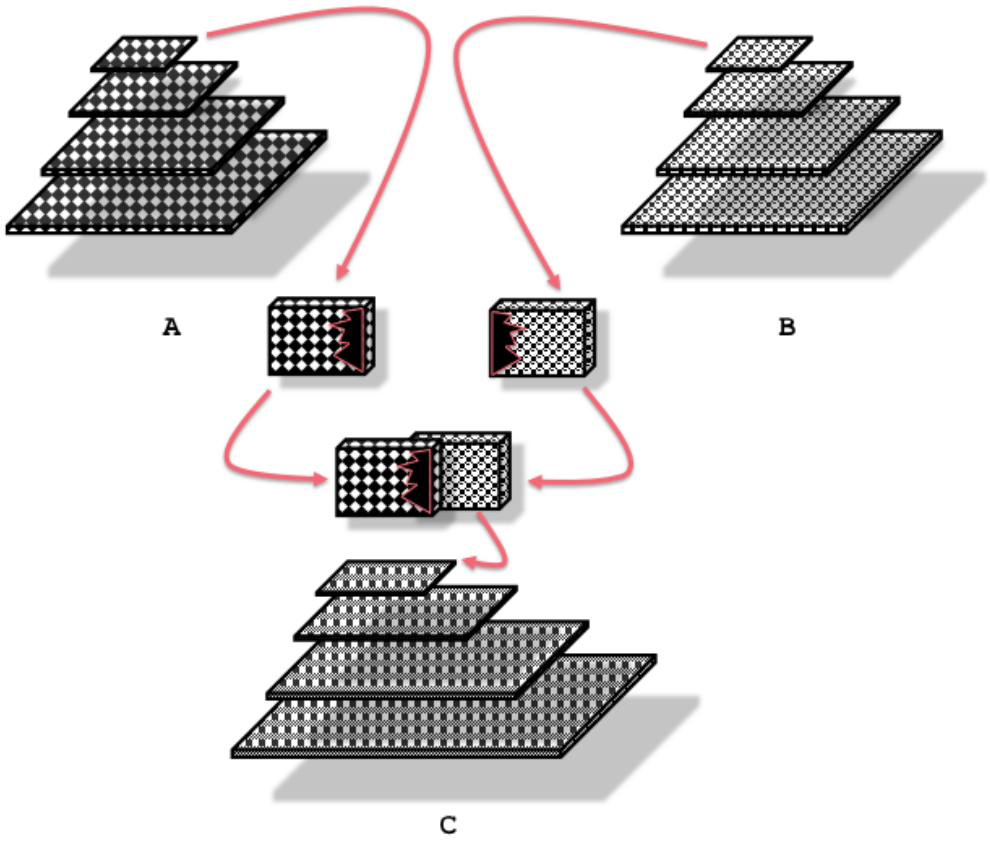


Figure 3.5: Laplacian Pyramid Blending Diagram

The choice of the stitching seam can be made through an analysis of the image, but to better show the accuracy of the results, this implementation choice to find the seam as a vertical line in the center of the overlap region. Joining each layer of the two Laplacian pyramids creates the panorama's Laplacian pyramid. In following the development of the Laplacian from the Gaussian, the RGB panorama can be found by upsampling and then summing each layer of the Laplacian pyramid in pairs, starting with the top layer. The exact reverse of the Gaussian to Laplacian operation. This Laplacian pyramid combination is thought of as combining the frequency content of the images and then rebuilding the panoramic image with the frequency content of the images together.

The pyramid generation is the most computationally complex portion of the blending, but given that it based on the same means of calculating the Gaussian pyramid reduction, it could ostensibly be made quite efficient by reuse of those same calculations. It is by no means an impassable limitation, but it has not been implemented in its most efficient method in the current algorithm.

CHAPTER 4: RESULTS

To evaluate the results of the algorithm they have been separated into three categories. The affine scenarios are the images from views that can be registered perfectly by purely affine homographies. In this category there are no realistic surveillance-type scenes available. To generate these unrealistic scenarios, cropped images coming from a larger image are used here. Using these ideal affine scenarios allows an evaluation of the algorithm under ideal conditions. Given that the algorithm performs an affine search, any affine views from a scene of modest entropy in their overlap, should result in perfect registration, as will be shown. A comparison to a manual registration is used to identify the implementation concerns and characteristics of the algorithm. Any affine cases in which the algorithm failed would not be due to parallax or occlusions, but to insufficient scene entropy or a failure in the feature generation. This category of results shows not only the theoretical accuracy of the algorithm, but was of great use during the testing phase of the implementation.

Unless all objects in the scene are at infinite or equivalent depth, multiple views will not be accurately relatable by an affine homography. The near-affine scenarios are those with minimal amounts of parallax and little-to-no occlusion. These are realistic views of real scenes, but under restricted conditions such as minimal camera-to-camera rotation, relatively consistent depth variation between objects in the scene, and/or large camera-to-scene distance compared to the object sizes. These cases test the appropriateness of the algorithm in realistic, but simplistic, scenarios. These were used as a stepping stone to the completely realistic scenes and show the accuracy of the affine homography as an estimate in simple projective scenarios (*i.e.* those devoid of considerable parallax or occlusions). Again, manual registration test cases are used (still using an affine homography) for benchmarking the implementation.

The last category is the projective views and complex scenes, which are the most irregular but also the most realistic scenarios. These are most appropriately modeled by a Fundamental matrix transformation or even a non-linear polynomial transformation, if at all. These scenes present parallax disparities between the views and the views are all subject to object or motion

occlusions. Manual registration testing, maintaining minimal computational complexity, found that a convincing view is often capable with a projective homography and as such these scenes are labeled as being projectively related. The parallax disparities and occlusions could present views that cannot be registered on a pixel-to-pixel basis, but our goal is to generate a convincing panoramic view, not to generate pixel or sub-pixel registration. Ideally the algorithm would have advanced to apply the WFMI metric in a projective search-space, but results will show that the accuracy allowable by the much more efficient affine search-space were sufficient for estimating convincing views. Initial attempts to extend to the projective search-space, as discussed in Chapter 3, were found to be too computationally expensive to implement.

4.1 Affine Views

The rooftop scene, Figure **??**, is a view from the Tufts University campus in Massachussetts. As mentioned, this is a large image that has been cropped into two overlapping views. By testing the WFMI algorithm with these two ideal views (and other similar scenarios) a better understanding of the feature generation, entropy concerns, and overlap limitations for the algorithm were found. Through empirical testing of affine views of realistic (but affine) scenes it was found that a minimum of 10% pixel overlap could be tolerated on average. As the entropy of the scene in the overlap region increases, more distinct and varied features can be generated, and the less overlap that is required between the views. This directly follows from the implementation of the WFMI metric which calculates similarity based on the distributions of features in the overlap regions. Scenes with large amounts of entropy will have very distinct features that will have a low probability of reoccurring elsewhere in the scene, and thus elsewhere in any of the views besides in the overlap region. In an extremely unrealistic, high entropy, case it was found that the WFMI algorithm can register views with only 1% of the total pixels overlapping. To provide a general metric, realistic affine views (views similar to surveillance views, such as the Rooftop views) of a purely affine scene were found to require a minimum of 10-15% of the total number of pixels be in the overlap region.

The automatically blended rooftop views in Figure 4.2 can be compared to the manually blended views in Figure 4.3. The manual blending technique follows the WFMI algorithm's multi-resolution spline blending algorithm but the affine homography is generated by MATLAB®'s feature correspondence selection tool (*cpselect*) and the correspondence to transform function (*cp2tform*). These views were produced with roughly 16% of the pixels from each view occurring in the overlap region. This is a relatively low entropy scene as there are very large areas of flat texture (concrete, glass, sky, trees, *etc.*).

As can be seen in Figure 4.3 there is a rotation disparity from the manual results and the stitching seam is clearly visible as the views have been misregistered. In order to attempt to compare the algorithm to the manual method in terms of practical implementation, the feature points for the manual transform derivation were chosen relatively quickly at the building and window corners. There were 12 to 15 points chosen and they were automatically passed into the least-squared algorithm in MATLAB®'s *cp2tform* function. The idea was that if a user could choose accurate points in a comparable amount of time for the computation of the automated registration then the automated registration could be compared to the current system in place by the grant provider. This is not a rigorous test, but it shows that even for an affine view, point-to-point correspondence has far less tolerance for error.

To show the aforementioned statistical errors that can occur and why the filtering operation is a crucial and useful addition to basic MMI algorithms, the views in Figure 4.5 were degraded by additive white Gaussian noise resulting in an average SNR between the two views of roughly 24 dB. The degradation is clearly visible and will surely affect the calculation of the color gradients. The maps in Figure 4.4 are the AFNMI (left) and NMI maps for the registration of the noisy images. Clearly there is a large peak in the NMI map towards the center, but a smaller sharper peak off to the left side, as shown. This smaller peak is the true translation, but it would go undetected in a normal MMI implementation. As shown in the AFNMI map, it is extracted as the true peak, far above any other possibility. These maps are shown normalized to the height of their peak, the actual scale is not visually useful and would have only added confusion.

An alternate affine set of views is presented in the stone wall scene. This is again a single view of a realistic scene with two overlapping affine views cropped from the larger view. This is a view of a wall and warehouse in Allston, MA. These views have only 10% of the total image pixels in the overlap and are registered perfectly by the automatic WFMI algorithm. While they seem to be a generally low entropy pair of images, they are extremely different in their content. Only in the overlap region is there a rectangular region that corresponds, based on the shadow and the stones present in the views. This allows a much smaller overlap, despite the lack of entropy, and where point correspondence may be easily confused by the low entropy metal siding of the building (repeated pattern), the WFMI algorithm is a structural search that is extremely robust in such cases.

Again to show the strength of the algorithm, noise has been added to the images, this time though the registration can stay accurate down to an average SNR of roughly 16 dB because of the uniqueness of the overlap region between the views.

Lastly, to show the accuracy of the affine case, a cropped scene of the Stone Wall was modified to only have 1% of the pixels in each view be part of the overlap region and the accuracy is maintained. Blending errors from the image padding as discussed in the previous chapter are visible near the top of the image as there is an increase in brightness with no apparent source. Again, that is an implementation artifact (as in, the choice of the means of implementation) not an error in the design or algorithm, and not an error in that it was implemented incorrectly.

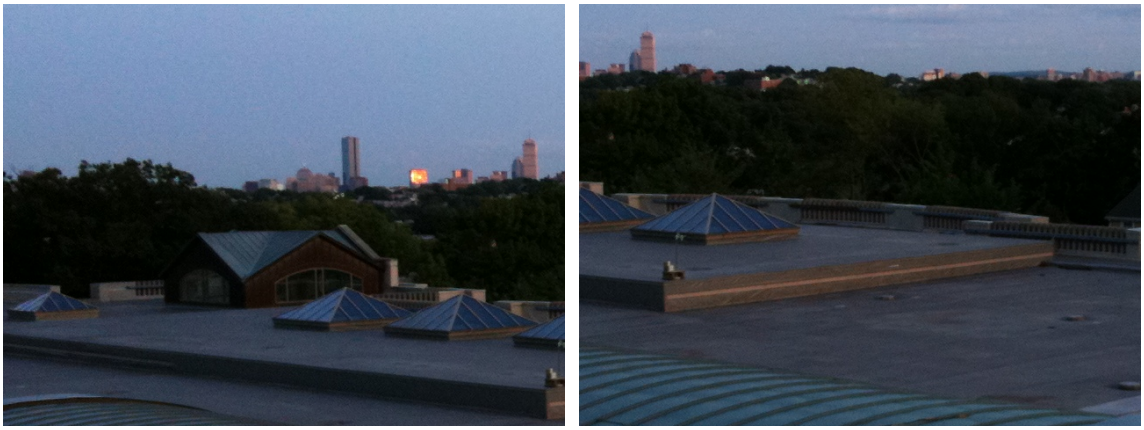


Figure 4.1: Rooftop Views (a) Left View, (b) Right View

Figure 4.2: Rooftop Views Blended

Figure 4.3: Rooftop Views Blended Manually (Affine)

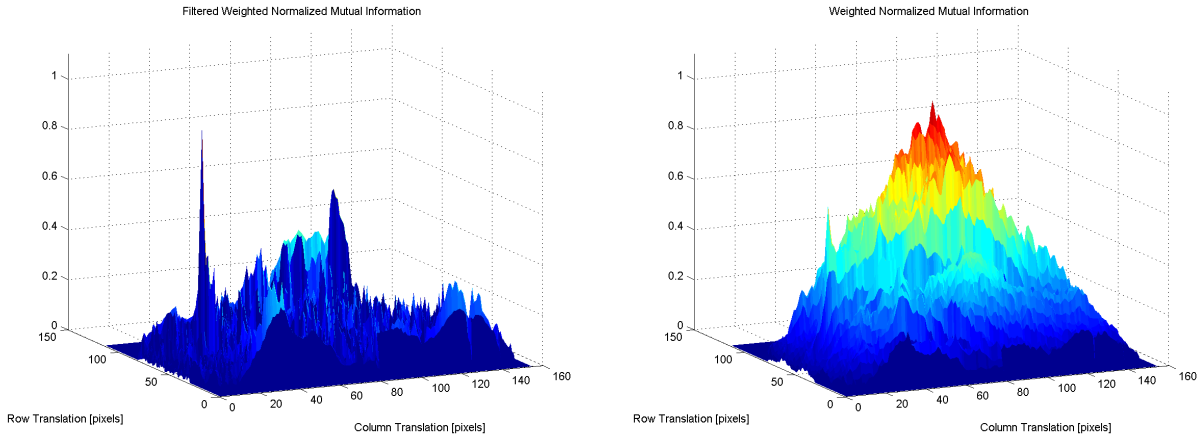


Figure 4.4: Mutual Information Maps from Translation Search (a) Filtered and Weighted, (b) Weighted

Figure 4.5: Rooftop Views with Average SNR of 24.122 dB (a) Left View, (b) Right View

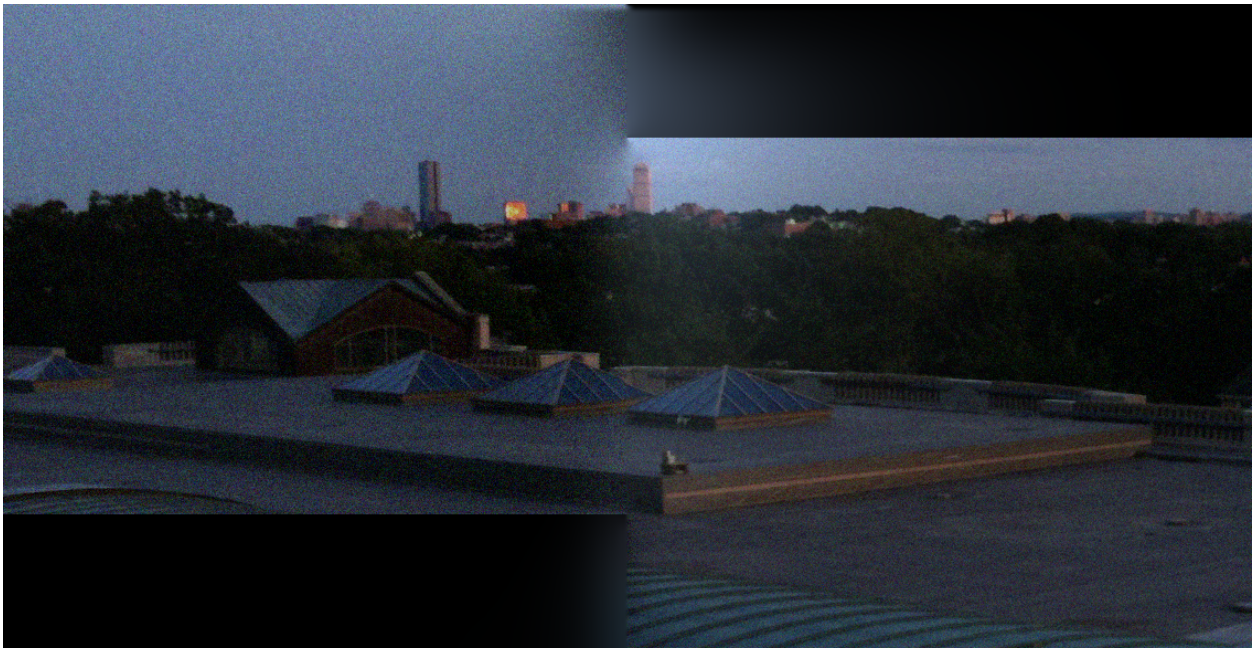


Figure 4.6: Rooftop Noisy Views Blended (Average SNR: 24.122 dB)

Figure 4.7: Stone Wall Scene (a) Left View, (b) Right View

Figure 4.8: Stone Wall Views Blended

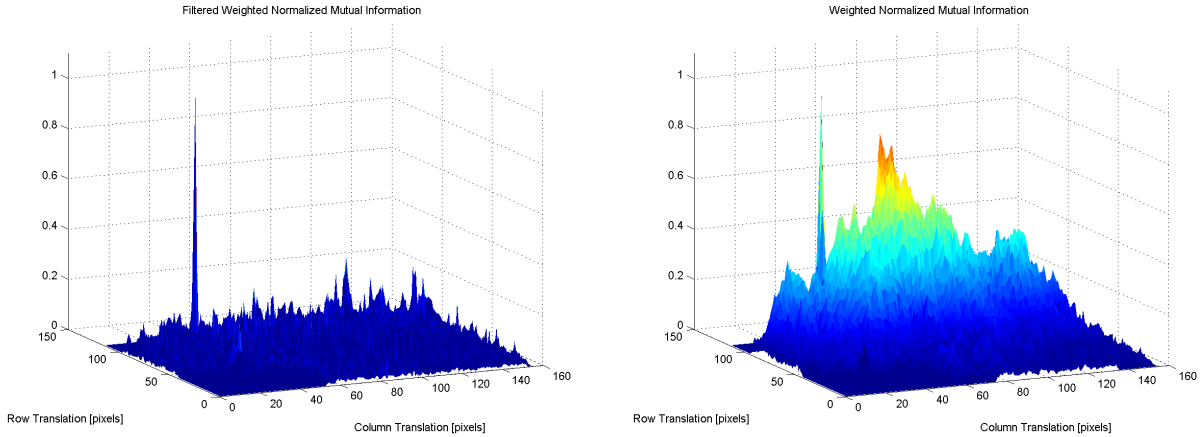


Figure 4.9: Mutual Information Maps from Translation Search (a) Filtered and Weighted, (b) Weighted

Figure 4.10: Stone Wall Views with Average SNR of 16.473 dB (a) Left View, (b) Right View

Figure 4.11: Stone Wall Noisy Views Blended (Average SNR: 16.473 dB)

Figure 4.12: Stone Wall Views Blended with 1% Overlap

4.2 Near-Affine Views

The affine related views of the previous section are entirely unrealistic, as stated, but in making a slow transition to evaluate the algorithm in realistic scenarios, views with near-affine relations can be generated in reality. In Figure 4.13 the views are not affine related, but being that there is only a slight rotation between the views (the right view's camera was rotated only a few degrees along the horizontal axis in-line with the camera centers) there is almost no occlusion or parallax disparity, and there is very little depth variation in the objects between the views. While all these characteristics (occlusion, parallax, and depth) do exist in the views, they are minimal enough that they can be ignored and an affine homography should perform quite well in registering the views.

Applying the WFMI algorithm to the views from Figure 4.13 produces the automatically generated panorama in Figure 4.14 with a manually generated panorama presented in Figure 4.15. A second set of views for the same scene is shown in Figure 4.16, with a third set of views in Figure 4.18. For these two sets, the difference is that the angle of rotation of the second camera was much larger than the previous view(s). This creates more and more depth distortion, and less and less overlap. Also, because of the nature of the scene, the more rotation there is, the less the objects overwhelm the scene structure and the repeated pattern (low entropy) of the brick floor begins to be the majority of the scene content. This makes registration extremely difficult, but again as the WFMI algorithm is structural in nature, it is shown in the two alternate sets of views that the registration still provides a very accurate estimate beyond the near-affine scenarios and in the presence of very low entropy.

These results show that the algorithm does function for real scenes with real scene content, but these views were generated under restrictions on parallax and occlusion, despite the robustness of the algorithm in the face of entropy and overlap concerns. The next section is the true test of the algorithm as all the views are not only susceptible to low entropy and minimal overlaps, but there is often extreme amounts of parallax disparity and occlusion variation for objects between the views.

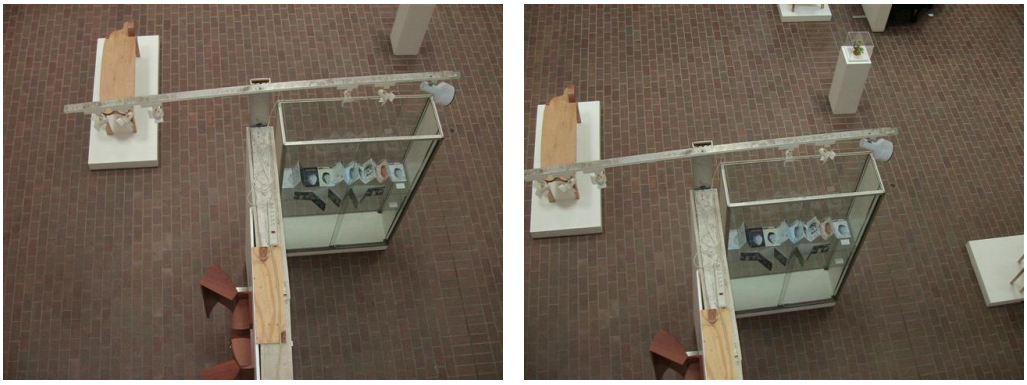


Figure 4.13: Art Gallery Scene (a) Left View, (b) Right View

Figure 4.14: Art Gallery Views Blended

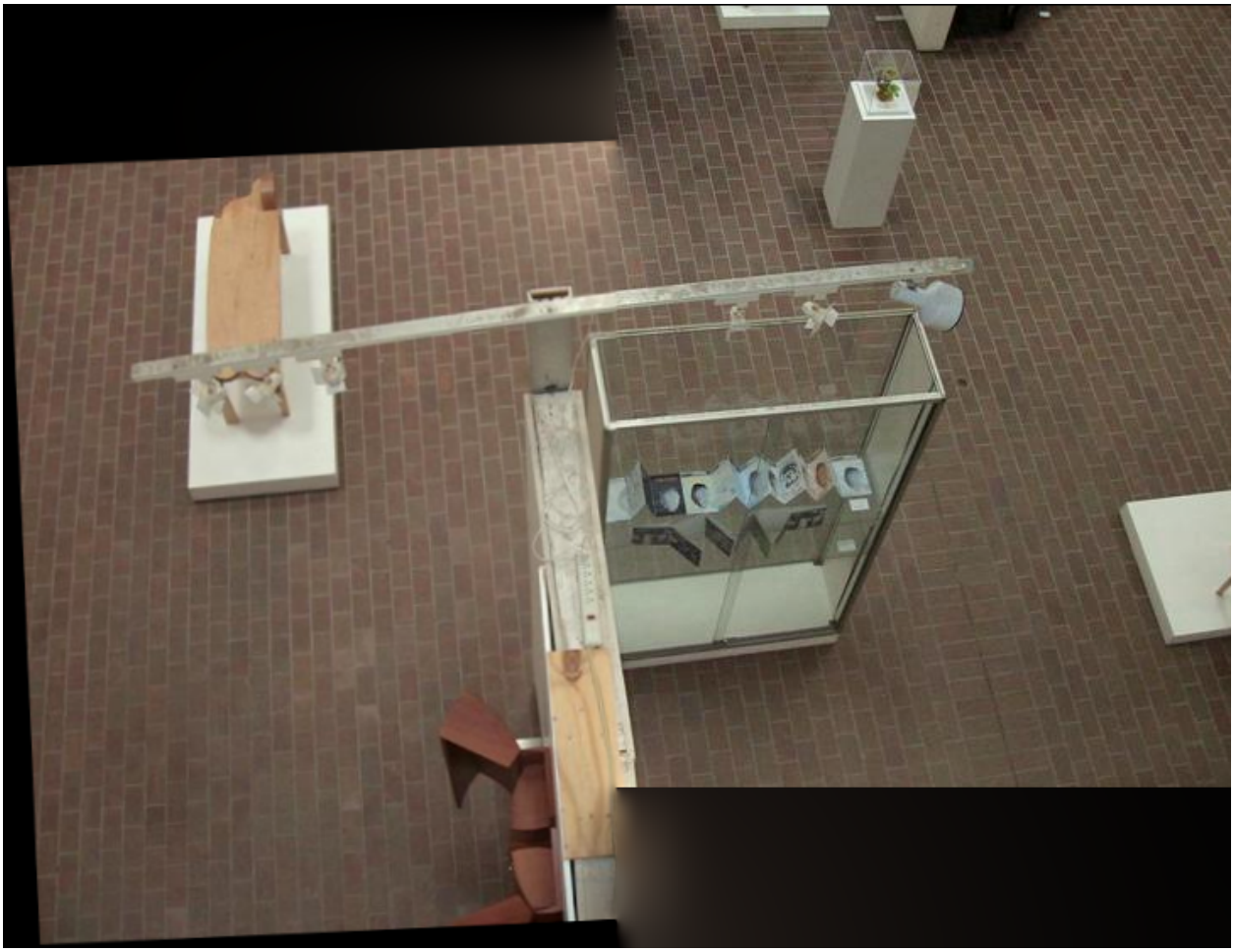


Figure 4.15: Art Gallery Views Blended Manually (Affine)

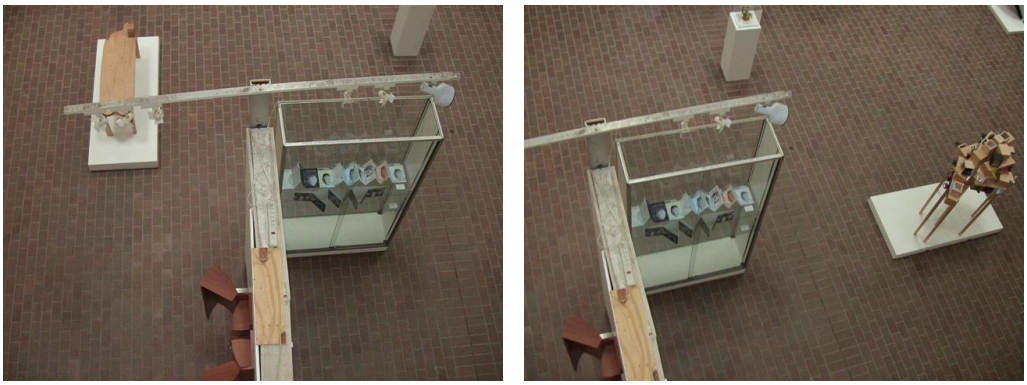


Figure 4.16: Art Gallery Scene (Modest Angle) Views (a) Left View, (b) Right View

Figure 4.17: Art Gallery (Modest Angle) Views Blended

Figure 4.18: Art Gallery (Large Angle) Views (a) Left View, (b) Right View

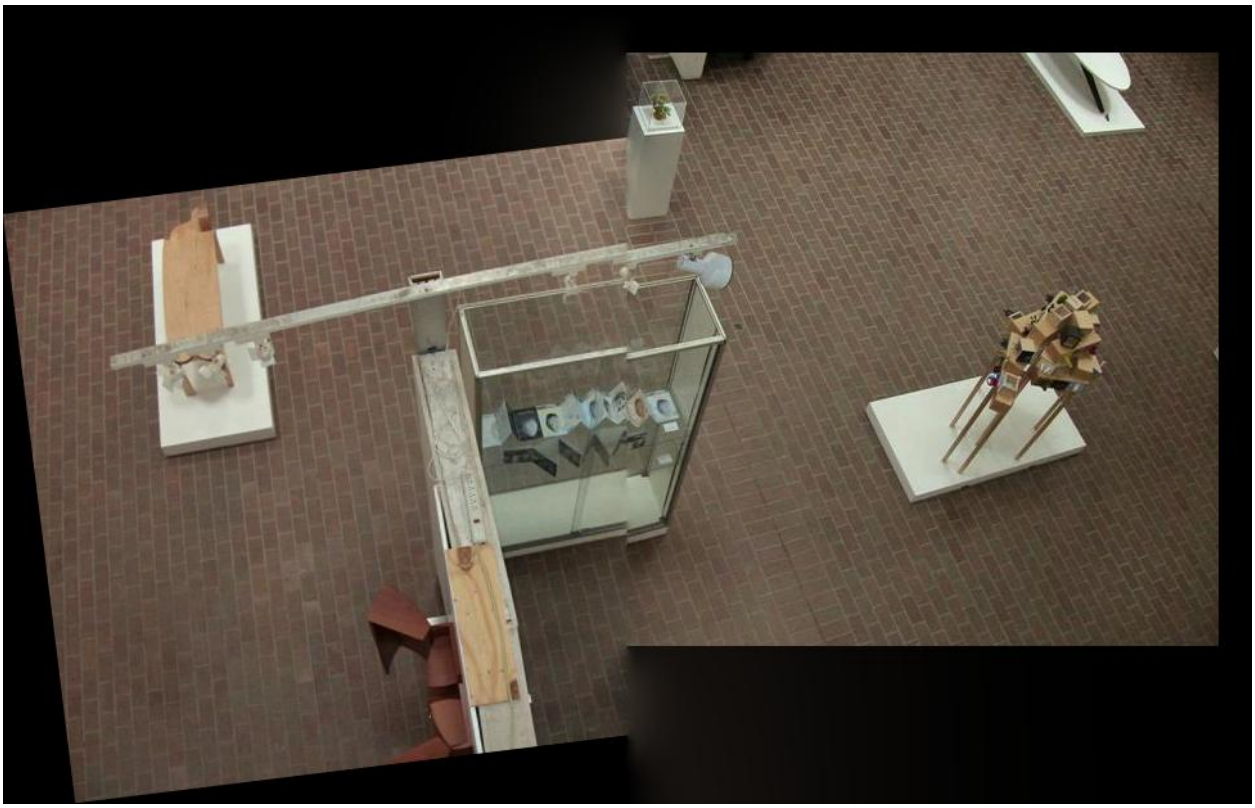


Figure 4.19: Art Gallery (Large Angle) Views Blended

4.3 Complex Views

This final set of results are all realistic scenarios for surveillance, most of which come from real, stationary security cameras. The real surveillance image sets are much poorer in quality as they are susceptible to all the previously discussed weather and illumination artifacts. The cameras are usually not as good quality as a commercial personal use camera (from which all the previous images were taken, as well as the sets in Figures 4.20 and 4.22). So these results show the strength of the algorithm not only for its use of a simple affine search in the face of occlusion and parallax disparities, but also in the face of real noise, real illumination variations, and truly unknown and uncontrolled camera parameters. As with all the results a very simple stitching seam (vertically halfway through the overlap region) was chosen and it is visible in all of the following results as none of these scenes are mathematically relatable by only an affine homography. However, all the results present an accurate estimate for the overlap region with no added complexity from the previous scenarios. These are all based on simple and fast calculations, that still have room for optimization to real time operation, and while "ghosting" and "jumps" are very visible, the views present a convincing panorama of the scene that is easily understood by a viewer. Especially given the color variations and depth discrepancies between the views, it can be a challenge (when unfamiliar with the scenes) to understand the true relationship between the views. These views, as all the previous ones, are generally of size 480×640 (row by column) and with all the implementation features mentioned in the previous chapter, they can be run through the entire algorithm (including being read in, stitched, and written out to file) in about 10 minutes for a modest search space in the MATLAB® implementation. The OpenCV implementation was not completed for the entire algorithm, but the core components (transformation search, registration, and blending) were implemented and ran in less than a minute with very modest hardware specifications in both methods (Intel® Core 2 Duo with 2GB RAM). There is room for improvement, but this is clearly a robust and useful metric for registration and has been presented as a novel solution to a very complex problem as these scenes have all the problems that are typically avoided by most registration algorithms and are often only tackled one at a time, rarely does it occur that an algorithm attempts

to overcome so many issues simultaneously with a combined effort and gestalt view of the problems and scenarios. That is the strength and core of this algorithm, that all the various problems come down to producing real, singular images; images of interest don't typically have one or two problems, they have them all, and so in designing an algorithm to overcome a problem it should take into consideration where and when that problem occurs, and often that will be alongside many other problems which cannot be ignored.

Manually registered views are not presented for comparison as it would be pointless to attempt to manually register these views with a clearly irrelevant affine homography. To show the views registered manually and accurately would also be pointless as it does not enhance an understanding of where the errors occur as they are all very visible, and the correct views would be distorted in a 3-D space, and thus irrelevant to compare to results restricted to the 2-D image planes.

Figure 4.20: Art Gallery Scene 01 (a) Left View, (b) Right View

Figure 4.21: Art Gallery 01 Views Blended

Figure 4.22: Art Gallery Scene 02 (a) Left View, (b) Right View

Figure 4.23: Art Gallery 02 Views Blended

Figure 4.24: Lenel Front Lot Scene (a) Left View, (b) Right View

Figure 4.25: Lenel Front Lot Views Blended

Figure 4.26: Lenel Back Lot Scene (a) Left View, (b) Right View

Figure 4.27: Lenel Back Lot Views Blended

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

The presented algorithm generates convincing panoramic views automatically for complex scenes. The research here has investigated the robustness of mutual information as a metric for complex, realistic scenes. There is still a large amount of room for improvement, but this work has shown the strength and versatility of mutual information, especially for overcoming difficult problems such as parallax differences and object occlusions between multiple views. In the case of occlusion it is important to keep in mind that there is no perfect registration as portions of the scene are available in one view but not in the other(s). In these cases, the best registration would be the one that can correspond the pixels of objects that exist in both views. This scenario is the prime case for the strength of the WFMI algorithm, since there is not a reliance on sparse pixel-to-pixel correspondences, but rather segment-to-segment correspondence. Therefore, occlusion is not a crippling problem for the WFMI registration algorithm. In the case of parallax differences between views and the complexities arising from the projective geometry of multi-view imaging, the appropriate homography may not exist and a Fundamental matrix or a non-linear polynomial mapping may be more appropriate. Categorizing these scenarios as Projective in nature (as in: related by a projective homography) was shown as an appropriate simplification in the manual registration examples, but the WFMI algorithm has shown that the even further simplification to the affine search-space can still produce accurate registration, or an appropriate estimate in the more complex scenarios.

5.1 Future Work

Originally this work was set out as a project to replace a system of manual image registration for surveillance videos by means of a fully automatic software implementation utilizing stationary cameras with unknown locations and no *a priori* information, besides allowing for the assumption of some amount of overlap between views that can be registered. The complexity of the scenario

and project guidelines went beyond the initial scope of the project, but the novel application of mutual information was introduced and tested. An initial expansion of the algorithm would be to finalize the registration for non-affine views by enhancing the algorithm with a pixel-to-pixel correspondence algorithm applied to the determined overlap region. Given that the WFMI algorithm provides intelligent estimates for the registration of views of complex scenes, relatively simple registration algorithms could be successful when applied to the estimated overlap region. In this respect the WFMI algorithm would be setting initial conditions to limit the search space for pixel-based registration, allowing for faster and more efficient computations in robust registration algorithms. In a practical implementation, the WFMI algorithm could provide an initial estimate on video feed frames and as more frames are calculated, refinements could be made to the initial estimate, especially as objects pass through the overlap regions of the views being registered.

There was also some initial research done in investigating the application of unsupervised image segmentation, such as the robust and accurate algorithm in [27], to generate better features for object correspondence. Again, since the WFMI algorithm is a region-based registration, the more intelligently that features are generated to define object boundaries rather than intensity data, the more accurate the mutual information metric will be, as overlapping segments containing the same object will be a relatively unique statistical event sharing the same feature histograms. However, this could greatly increase computational complexity as the segmentation would need to be robust considering the practical scenarios: uncontrolled weather, illumination, and camera artifacts.

In terms of advancing the application of the weighted and filtered mutual information metric itself, there is a lot of contemporary research moving towards scene understanding and 3-dimensional (spatial) image and video data. Multiple views of a scene directly allow for the extension of the projective geometry to develop 3-D information about the structure of the scene, and the objects within it. If a scene's structure is unknown from its views and an accurate registration is not available, the WFMI metric could be applied to identify objects, regions, or even elements of the scene in motion that are correlated between the views. Object tracking, depth reconstruction, and motion estimation could all be rich areas of research for this novel application of information

theory.

5.2 Final Remarks

Again, the work completed here was initially attempting to create a fully automatic panorama creation algorithm from overlapping views of an unknown scene at unknown locations with unknown cameras. While this initial goal cannot be described as being completely met, what has been produced is a novel application of mutual information as a metric in image registration, as it has extended beyond well-defined affine scenarios with large amounts of overlap in well-conditioned views of simple scenes. It is a robust and accurate algorithm for affine views and is an intelligent estimate when extended to projective views of complex, realistic scenes. This work shows the beginnings of extending the known robustness of mutual information as a metric for image and video data correspondence. There are many avenues in which to extend this research that progress in line with modern research as digital imagery is being view more and more as holding extensive amounts of information beyond the reach of basic intensity and gradient calculations. There is also no limitation to the applications where the WFMI algorithm can be used in conjunction with point-to-point algorithms to find the true homography between views. Especially considering that this is video data, the WFMI algorithm can be applied to the first frames and a much simpler point-to-point algorithm can take advantage of these initial conditions and present a completely accurate panorama. Coupled with the potential for real-time operation, the WFMI algorithm can quickly and efficiently create an accurate registration estimate, that can then be immediately adapted to be more accurate and present a viewer with the most accurate result without any knowledge of the scene and by avoiding the complex computations of point-to-point algorithms.

REFERENCES

[1] Barbara Zitová and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977 – 1000, 2003.

[2] M. Brown and D. G. Lowe. Recognising panoramas. pages 1218–, 2003.

[3] Michael Brown, Richard Szelisk, and Simon Winder. Multi-image matching using multi-scale oriented patches. 1:510–517, 2005.

[4] 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, 2003.

[5] Paul Viola and William M. Wells III. Alignment by maximization of mutual information. 24(2):137–154, 1997.

[6] Karl C. Walli, David R. Nilosek, John R. Schott, and Carl Salvaggio. Airborne synthetic scene generation (aerosynth). In *Proceedings of the ASPRS, ASPRS/MAPPS 2009 Fall Conference, Digital Mapping - From Elevation to Information, Digital Elevation Data Fusion Innovations*, San Antonio, Texas, United States, November 2009. ASPRS.

[7] Alex Rav-Acha, Yael Pritch, Dani Lischinski, and Shmuel Peleg. Dynamosaics: Video mosaics with non-chronological time. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:58–65, 2005.

[8] David Nilosek and Karl Walli. Aerial scene synthesis from images. In *SIGGRAPH '09: Posters*, SIGGRAPH '09, pages 65:1–65:1, New York, NY, USA, 2009. ACM.

[9] Yasushi Kanazawa and Kenichi Kanatani. Image mosaicing by stratified matching. *Image and Vision Computing*, 22(2):93 – 103, 2004. Statistical Methods in Video Processing.

[10] Thomas Haenselmann, Marcel Busse, Stephan Kopf, Thomas King, and Wolfgang Effelsberg. Multi perspective panoramic imaging. *Image and Vision Computing*, 27(4):391 – 401, 2009.

[11] Nuno Gracias, Mohammad Mahoor, Shahriar Negahdaripour, and Arthur Gleason. Fast image blending using watersheds and graph cuts. *Image and Vision Computing*, 27(5):597 – 607, 2009. The 17th British Machine Vision Conference (BMVC 2006).

[12] Matthew Brown and David Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74:59–73, 2007. 10.1007/s11263-006-0002-3.

[13] Xiaofeng Fan, Harvey Rhody, and Eli Saber. An algorithm for automated registration of maps and images based on feature detection and mutual information. 6813(68130F), 2008.

[14] Chris Harris and Michael Stephens. A combined corner and edge detector. In *Proceedings of the Fourth Alvey Vision Conference*, pages 147–151, 1988.

[15] O. Faugeras, Q.T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. MIT Press, 2004.

[16] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[17] Peter J. Burt and Edward H. Adelson. A multiresolution spline with application to image mosaics. *ACM Trans. Graph.*, 2:217–236, October 1983.

[18] Noah Snavely, Steven Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80:189–210, 2008. 10.1007/s11263-007-0107-3.

[19] S.E. Palmer. *Vision science: photons to phenomenology*. A Bradford book. MIT Press, 1999.

[20] A. Papoulis and S.U. Pillai. *Probability, random variables, and stochastic processes*. McGraw-Hill electrical and electronic engineering series. McGraw-Hill, 2002.

[21] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, Upper Saddle River, NJ, 3rd edition, 2008.

[22] F. M. Reza. *An Introduction to Information Theory*. Dover Publications, New York, NY, 1994.

[23] D.J.C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2004.

[24] S. Kullback. *Information theory and statistics*. Dover books on mathematics. Dover Publications, 1997.

[25] T.M. Cover and J.A. Thomas. *Elements of information theory*. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience, 2006.

[26] Y. Yao, S. Wong, and C. Butz. On information-theoretic measures of attribute importance. In Ning Zhong and Lizhu Zhou, editors, *Methodologies for Knowledge Discovery and Data Mining*, volume 1574 of *Lecture Notes in Computer Science*, pages 133–137. Springer Berlin / Heidelberg, 1999.

[27] L. Garcia Ugarriza, E. Saber, S.R. Vantaram, V. Amuso, M. Shaw, and R. Bhaskar. Automatic image segmentation by dynamic region growth and multiresolution merging. *Image Processing, IEEE Transactions on*, 18(10):2275 –2288, 2009.

[28] Hsien-Che Lee and David R. Cok. Detecting boundaries in a vector field. 39(5):1181–1194, 1991.

[29] D. Schneider, E. Schwalbe, and H.-G. Maas. Validation of geometric models for fisheye lenses. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(3):259 – 266, 2009. Theme Issue: Image Analysis and Image Engineering in Close Range Photogrammetry.

[30] Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. pages 251–258, 1997.

[31] J. Davis. Mosaics of scenes with moving objects. pages 354 –360, June 1998.

[32] A.V. Oppenheim, R.W. Schafer, and J.R. Buck. *Discrete-time signal processing*. Prentice-Hall signal processing series. Prentice Hall, 1999.

[33] J.S. Lim. *Two-dimensional signal and image processing*. Prentice-Hall signal processing series. Prentice Hall, 1990.

[34] A. Murat Tekalp. *Digital Video Processing*. Prentice Hall, Upper Saddle River, NJ, 1995.

[35] Danial Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. 47(1-3):7–42, 2002.