

Air Force Institute of Technology

AFIT Scholar

Theses and Dissertations

Student Graduate Works

3-2003

Feature Guided Image Registration Applied to Phase and Wavelet-Base Optic Flow

Kate R. Duffy

Follow this and additional works at: <https://scholar.afit.edu/etd>



Part of the [Applied Mathematics Commons](#), and the [Signal Processing Commons](#)

Recommended Citation

Duffy, Kate R., "Feature Guided Image Registration Applied to Phase and Wavelet-Base Optic Flow" (2003). *Theses and Dissertations*. 4239.
<https://scholar.afit.edu/etd/4239>

This Thesis is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact richard.mansfield@afit.edu.



**FEATURE GUIDED REGISTRATION
APPLIED TO PHASE- AND WAVELET-BASED OPTIC FLOW
ALGORITHMS**

THESIS

Kate R. Duffy, Second Lieutenant, USAF

AFIT/GE/ENG/03-09

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

**AIR FORCE INSTITUTE OF
TECHNOLOGY**

Wright Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/03-09

FEATURE GUIDED IMAGE REGISTRATION
APPLIED TO PHASE- AND WAVELET-BASED OPTIC FLOW
ALGORITHMS

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Electrical Engineering

Kate R. Duffy, B.S.E.E. Northeastern University
Second Lieutenant, USAF

March 2003

Approved for public release; distribution unlimited

AFIT/GE/ENG/03-09

FEATURE GUIDED IMAGE REGISTRATION
APPLIED TO PHASE- AND WAVELET-BASED OPTIC FLOW
ALGORITHMS

Kate R. Duffy, B.S.E.E. Northeastern University

Second Lieutenant, USAF

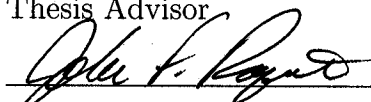
Approved:



Maj Roger L. Claypoole Jr. Ph.D.
Thesis Advisor

14 MAR 03

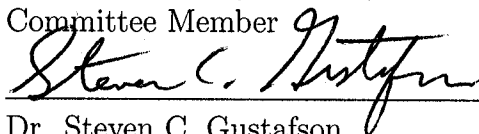
Date



Maj John F. Raquet Ph.D.
Committee Member

14 MAR 03

Date



Dr. Steven C. Gustafson
Committee Member

17 MAR 03

Date

Acknowledgements

I would like to thank Major Claypoole, Major Raquet, and Dr. Gustafson for helping with my thesis. I would specifically like to thank Major Claypoole, my thesis advisor, for helping me create the ‘best read’ document I’ve ever written. Thanks for putting in all those extras hours for me this quarter and throughout the year. I also thank Major Raquet for his navigational expertise. His help with interpretation of GPS data was crucial.

Finally, I would like to thank my friends and family for helping me get through my time at AFIT. Thanks Monica and Brendan for always being there. I especially want to thank my parents for always giving me the words of advice and confidence that I needed. Thanks to my brothers, sister and cousins who always helped me blow off steam when I came home for a visit. Finally, thanks to the Luongo family for always being there.

Kate R. Duffy

Table of Contents

	Page
Acknowledgements	iii
List of Figures	vii
List of Tables	ix
Abstract	x
I. Introduction	1-1
1.1 Problem Statement	1-1
1.2 Scope	1-2
1.3 Thesis Organization	1-2
II. Background	2-1
2.1 Introduction	2-1
2.2 Background of Our Problem	2-1
2.3 Image Registration	2-1
2.3.1 Components of Image Registration	2-2
2.3.2 Image Registration Techniques	2-3
2.4 Wavelet Transform	2-4
2.5 Morphology	2-8
2.6 Optic Flow Overview	2-8
2.7 Optic Flow Techniques	2-11
2.7.1 Phase-Based Optic Flow Technique	2-11
2.7.2 Wavelet-Based Optic Flow	2-13
2.8 Calculate the Full Velocity	2-13
2.9 Summary	2-14

	Page
III. Methodology	3-1
3.1 Introduction	3-1
3.2 Wavelet-Based Feature Extraction	3-2
3.2.1 Pre-Filtering the Data Imagery	3-2
3.2.2 Selection of Wavelets Basis for Feature Extraction	3-3
3.2.3 Shift-Invariant Wavelet Transform	3-3
3.2.4 Subband Choices	3-5
3.2.5 Lowpass Filter	3-6
3.2.6 Threshold and Binary Masking	3-6
3.2.7 Morphological Imaging	3-7
3.2.8 Creating Regions of Interest (ROI)	3-9
3.2.9 Additional Parameters that Modify our Code.	3-11
3.3 Wavelet-Based Translation Algorithm	3-14
3.3.1 Number of Significant Coefficients	3-14
3.3.2 RDWT	3-14
3.3.3 Masking	3-14
3.3.4 Comparison of Location	3-14
3.3.5 Correlations	3-15
3.4 Phase-Base Optic Flow Algorithm	3-15
3.4.1 Methodology for Optic Flow Algorithm	3-15
3.5 Velocity Computation	3-16
3.5.1 Methodology for Altitude Calculation	3-17
3.5.2 Velocity calculation	3-17
3.6 Summary	3-18
IV. Results	4-1
4.1 Introduction	4-1
4.2 Discussion of Validation Study	4-1

	Page
4.2.1 Test Images	4-1
4.2.2 Peak Signal-to-Noise Ratio	4-2
4.3 Experiment	4-4
4.3.1 Synthetic Data	4-4
4.3.2 Video Stream Data Analysis	4-10
4.4 Summary	4-19
V. Discussion and Future Work	5-1
5.1 Contributions of this Thesis	5-1
5.2 Potential for Future Research	5-1
5.2.1 Other Image Registration Algorithms	5-1
5.2.2 Use of Subbands	5-2
5.2.3 Finding Redundant Objects	5-2
5.2.4 Dealing with Video Stream Data Rotations	5-3
Bibliography	BIB-1
Vita	VITA-1

List of Figures

Figure		Page
2.1.	<i>Filter Bank Representation of the Discrete Wavelet Transform.</i>	2-4
2.2.	<i>The Discrete Wavelet Transforms of a Video Image.</i>	2-5
2.3.	<i>Three Iterations of the Discrete Wavelet Transforms.</i>	2-6
2.4.	<i>The Discrete Wavelet Transform and the Redundant Discrete Wavelet Transform.</i>	2-7
2.5.	<i>Motion Field.</i>	2-9
2.6.	<i>Motion Field with Assumptions.</i>	2-10
2.7.	<i>Output Responses of a Gabor Filter.</i>	2-12
3.1.	<i>Original Video Image and Filtered Video Image</i>	3-3
3.2.	<i>Filter Bank Representation of the Redundant Discrete Wavelet Transform.</i>	3-5
3.3.	<i>The Masked and Thresholded Dominant Features.</i>	3-7
3.4.	<i>Morphological Shape.</i>	3-8
3.5.	<i>Subimages of the Video Image.</i>	3-10
3.6.	<i>Example of Regions of Interest on two sequential Images . . .</i>	3-11
3.7.	<i>Ugly and Bad Examples</i>	3-13
3.8.	<i>Optic Flow Vectors</i>	3-16
4.1.	<i>The Test Images.</i>	4-2
4.2.	<i>Examples of PSNR values for the Lenna image.</i>	4-3
4.3.	<i>Evaluation of Accuracy pixel shift of two with PSNR of 28 dB.</i>	4-5
4.4.	<i>Evaluation of Accuracy of pixel shift of two with PSNR of 22.</i>	4-6
4.5.	<i>Displacement Performance of Integer Values.</i>	4-8
4.6.	<i>Estimated Velocities of the Video Stream Data</i>	4-11
4.7.	<i>Estimated Velocities of the Video Stream Data for one second.</i>	4-12

Figure		Page
4.8.	<i>Video Stream Data of Invalid Estimates</i>	4-16
4.9.	<i>Corrected Velocities of Invalid Estimates</i>	4-17
4.10.	<i>Corrected Velocities</i>	4-18

List of Tables

Table		Page
3.1.	<i>Features per Wavelet Decomposition</i>	3-4
3.2.	<i>Features per RDWT Iteration.</i>	3-5
3.3.	<i>Feature Extraction Characteristics.</i>	3-12
4.1.	<i>Statistics of the Accuracy and Noise Graphs.</i>	4-7
4.2.	<i>Displacement Graphs Statistics.</i>	4-9
4.3.	<i>Valid Estimate Statistics.</i>	4-13
4.4.	<i>Statistics of Individual Features.</i>	4-14
4.5.	<i>Invalid Estimate Statistics.</i>	4-15
4.6.	<i>Invalid Data Corrected Statistics.</i>	4-16

Abstract

Optic Flow algorithms are useful in problems such as computer vision, navigational systems, and robotics. However, current algorithms are computationally expensive or lack the accuracy to be effective compared with traditional navigation systems. Recently, lower accuracy inertial navigation systems (INS) based on Micro-electromechanical systems (MEMS) technology have been proposed to replace more accurate traditional navigation systems. An Optic Flow algorithm can be created that is, unlike GPS, not susceptible to jamming or spoofing. Our long term goal is to use a robust Optic Flow algorithm in conjunction with a MEMS INS to create a more accurate and less expensive MEMS INS navigational system.

We propose a new wavelet-based technique for motion analysis based on feature extraction. With the use of the redundant discrete wavelet transform and morphological processing, dominate image features are efficiently extracted from the image, allowing for a robust and computationally effective optic flow algorithm. These features are used to guide the registration; the image is broken into subsections based on the spatial location and extent of the detected features. These subsections are incorporated into optic flow algorithms to create local estimates of image motion. We verify the accuracy of our algorithm by analyzing actual aircraft video with known position and inertial references.

FEATURE GUIDED IMAGE REGISTRATION APPLIED TO PHASE- AND WAVELET-BASED OPTIC FLOW ALGORITHMS

I. Introduction

1.1 Problem Statement

Image registration is used in a variety of applications such as computer vision, pattern recognition, and target location and definition. It is the process of taking two or more images and determining the best correlation between them to find the translation, rotation, or spatial characteristics of the object.

The military has a need to process large quantities of image data. We look at a specific application in support of the Munitions Directorate of the Air Force Research Lab (AFRL) at Eglin AFB, FL. Current Microelectromechanical systems (MEMS) inertial navigation system (INS) technology lacks the performance of conventional aircraft INS. One way to improve MEMS INS performance is to use image data of ground scenes. This method involves analyzing topographical data to identify and track objects and estimate their relationship to the weapon platform. It enables the system to calculate parameters needed to assist the INS. This thesis focuses on algorithms for object recognition, tracking, and estimation from all viewpoints. The long term goal of our research is to obtain an accurate, robust, and efficient algorithm that registers images and then calculates the velocities and the displacements of objects within the image. Various optic flow algorithms have been developed to fit this need, yet none have met all the requirements.

This research will allow AFRL at Eglin AFB to realize the potential of ground scene images for MEMS INS as opposed to the poor-performing MEMS gyros cur-

rently in use. If found to be a viable solution, our research will be used for countering electronic warfare measures such as jamming or spoofing of the global positioning navigation system (GPS).

1.2 Scope

Previous optic flow algorithms either lacked the accuracy or computational efficiency necessary for a real time navigation system. We analyze more accurate techniques and try to implement them in a less computationally intensive manner. This analysis is done with wavelet-based feature extraction.

Our research demonstrates the robustness, accuracy, and efficiency of our algorithm by application to synthetic and video stream data. The strengths and weakness will be explored so that we may gain a better understanding of where improvements need to be made. Finally, future research efforts are discussed.

1.3 Thesis Organization

Chapter Two introduces the basic theory of the concepts we use in the development of our algorithm. First, an overview of image registration and the criteria for accurate registration are provided. We then discuss why wavelets meet these criteria. The basic theory of wavelets is discussed along with how the redundant discrete wavelet transform and morphological processing best fit our needs.

Chapter Two also gives an overview of optic flow and how displacements can be calculated with such a technique. A brief overview of different approaches is provided. We go into further detail on the Wavelet- and Phase-Based Optic Flow algorithm, which are used in this thesis. A brief overview of Gabor filters is given, since they are used in the design of one of the tested algorithms.

Chapter Three begins with an overview of how we use wavelets in our feature extraction process. First, we analyze the redundant discrete wavelet transform at various scales to ensure persistency of our features. Next, we develop thresholding

and masking criteria for the wavelet subbands. We explain how the various subbands are combined with morphological processing to create and characterize objects. Regions of interest are created based on these objects. The image is then broken into subimages based on the regions of interest. These subimages become the inputs for the two algorithms we validate.

Chapter Three continues with the methodology of both Phase- and Wavelet-Based Optic Flow algorithms, along with specific parameters needed for valid estimates. Finally we discuss the velocity calculations and the components necessary for such calculations.

Chapter Four describes our validation study and how our Feature Guided algorithm performs against the original Phase- and Wavelet-Based Optic Flow algorithms. Synthetic data with additive white Gaussian noise along with an actual video stream sequence are used.

Finally, in Chapter Five we discuss our contributions along with recommendations for future research efforts.

II. Background

2.1 Introduction

This thesis uses wavelets and morphology to extract features for image registration and motion analysis. Thus, we begin our discussion with an overview of the problem. We discuss the basic components of image registration and how wavelets are the best fit for our purposes. We then consider the specific properties of wavelets and morphological processing that are useful in our algorithm. Next, we present the basic theory of optic flow. We show our reasoning for using phase- and wavelet-based optic flow algorithms and explain the fundamentals behind such algorithms.

2.2 Background of Our Problem

Current Microelectromechanical systems (MEMS) inertial navigation system (INS) technology lacks the performance of conventional aircraft INS. One way to improve INS performance is to use image data of ground scenes. This method involves analyzing topographical data to identify and track objects and estimate the relationship of these objects to the weapon platform. This procedure enables the system to calculate parameters needed to assist the INS. A key aspect of this thesis focuses on algorithms for object recognition, tracking, and estimation from all viewpoints. The long term goal of our research is to develop an accurate, robust, and efficient algorithm that registers images and can then calculate the velocities and the displacements of objects within the image. The first step in reaching this goal is to ensure that proper image registration takes place.

2.3 Image Registration

Image registration can be defined as the process in which two or more images, acquired from one or more sensors, are compared to calculate differences in rotation, translation, scalar properties, etc. For our purposes, we refer to the original image as

the reference image and the second image as the image or the sequential image. The primary difference between the two images is translation. Clearly, proper registration is a required first step for any comparison-based image processing.

2.3.1 Components of Image Registration. Brown [4] cites four major components necessary for proper image registration. They are: feature space, search space, the search strategy, and the similarity metric.

1. Feature Space

The feature space contains the characteristics that are extracted from the reference and sequential images to calculate the differences in the sequence. Traditional characteristics include edges, contours, intersections of lines, etc. It is imperative that the feature space be well defined so that proper image registration can take place [24]. A well defined feature space should have robustness and generality [24]. Robustness is needed so that the same feature is extracted regardless of warping that may occur in consecutive images. Warping can be caused by noise, translation, rotation, etc. Generally, it is also important to ensure that even if our algorithm is used for different applications, such as computer vision, navigational data, etc., we are able to extract dominant features. Li and Zhou [20] have proposed additional criteria for selecting proper features. Their primary criterion is consistency. Consistency is defined in four parts: features are at the same locations in each image, located in high contrast regions, proportionally distributed throughout the image, and are unique in their surrounding areas.

2. Search Space

The search space contains the set of potential transformations that define the correspondence between the two images. Three common transformations are rigid body, affine, and polynomial. Throughout this thesis we work on a data set that uses rigid body search spaces composed of translation, scaling, and

rotation of the input image(s). In this research, we are particularly interested in translation.

3. Search Strategy

The search strategy determines the set of allowable deformations. If the images are expected to translate, then the search strategy is to compare the image(s) with all possible translations of the reference [4].

4. Similarity Metric

The similarity metric is used to determine how well the reference and the image(s) match for the chosen search space. Correlation is one of the most common metrics [4].

2.3.2 Image Registration Techniques. Previous work has shown that wavelet-based techniques for image registration are better suited than Fourier-based techniques [1, 4, 5, 6, 15, 16, 17, 18, 19, 20, 23, 26]. There are several properties that make wavelets a natural choice for our image registration algorithm. Traditionally, image analysts have manually analyzed multiple image sequences to extract the differences of persistent objects. With the use of the wavelet transform, the maxima of the wavelet coefficients can be computed automatically [15]. It has been documented that the correlation between wavelet coefficients and a set of images is greater than the correlation between the images themselves, which makes wavelets a more accurate transform than previous methods [29]. Locality allows the wavelet transform to simultaneously convey frequency and time information, unlike the Fourier-based methods which carry exact frequency information, but with spatial information entirely embedded in the phase. Finally, persistence is the last property important for our algorithm. True features are persistent across scales of the wavelet transform. This causes coefficients of large magnitude to be at the same spatial location in each scale. Clearly, because of these properties, wavelets are well suited for registration. Thus, we present a brief background on the wavelet transform.

2.4 Wavelet Transform

The wavelet transform, or more specifically the two-dimensional discrete wavelet transform, is the basis for our feature extraction in the image registration process. It is necessary to use the two-dimensional discrete wavelet transform because we work with two-dimensional images. The discrete wavelet transform can be implemented by using cascading filters. Figure 2.1 shows the filter bank implementation of the one-dimensional discrete wavelet transform for three iterations. The $H(z)$ represents the lowpass filter while the $G(z)$ represents the highpass filter. A downsampling of two takes place after each of the filters. The two-dimensional discrete wavelet transform is performed in a separable fashion: the one-dimensional discrete wavelet transform is applied to each row of the image, then to each column.

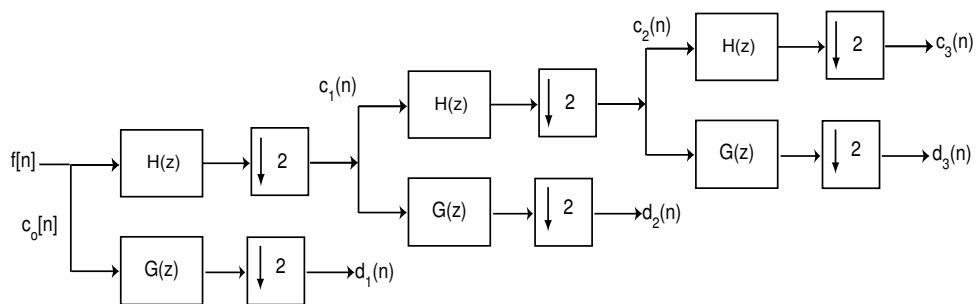


Figure 2.1. *Filter Bank Representation of the Discrete Wavelet Transform.*

The purpose of the two dimensional discrete wavelet transform is to dyadically decompose the image into four subbands. Each subband is one fourth the size of the original image; a downsizing of two along the rows and two along the columns takes place. Each of the subbands preserves certain characteristics of the original image. For example, see Figure 2.2 (a) for an original image and (b) the wavelet transform of that image. We use the following notation to define the subbands. The first letter stands for the type of filter applied to the rows, the second letter refers to the filter that is applied along the columns. L stands for a lowpass filter, and H stands for a highpass filter. The upper left is the Low-Low (LL) subband of the image, which forms a coarse approximation to the original image. The upper right

is the High-Low (HL) subband, which preserves the vertical edges. The bottom left is the Low-High (LH) subband, which preserves the horizontal edges. Finally the bottom right is the High-High (HH) subband, which preserves the 45 degree edges. These four subbands are easily distinguished in Figure 2.2(b) and Figure 2.3. Also, seen in Figure 2.3 are the multiple scales of the discrete wavelet transform, which are formed by iterating on the LL subband of the previous scale.

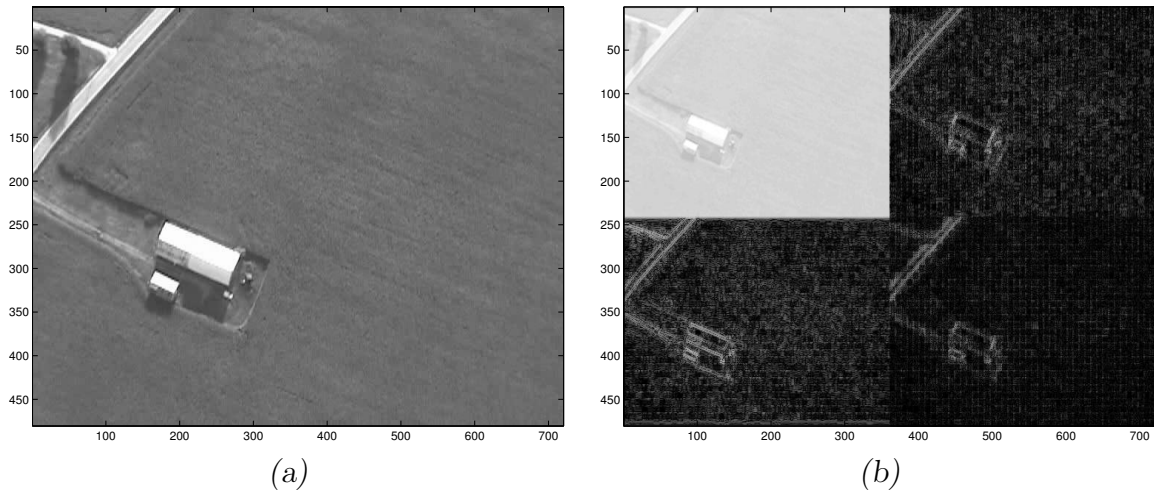


Figure 2.2. *Discrete Wavelet Transforms. (a) The Original Image. (b) One Iteration of the Discrete Wavelet Transform.*

For the purpose of our feature extraction, we take more than one iteration of the two-dimensional discrete wavelet transform to ensure robustness of the features, since significant features persist across scales. However, there is a point of diminishing returns. As more iterations are taken we decompose a coarser image each time. It can also be seen in Figure 2.3 that the subbands decrease in size with each decomposition. At some point the image becomes too coarse for reliable information extraction.

The two-dimensional discrete wavelet transform is created using downsampling. This causes the transform to be one-to-one, but shift varying. A shift invariant transform is crucial for algorithms that estimate translation. For this reason we chose the redundant discrete wavelet transform (RDWT), which still decomposes the image into various scales, yet it is shift invariant and each subband preserves

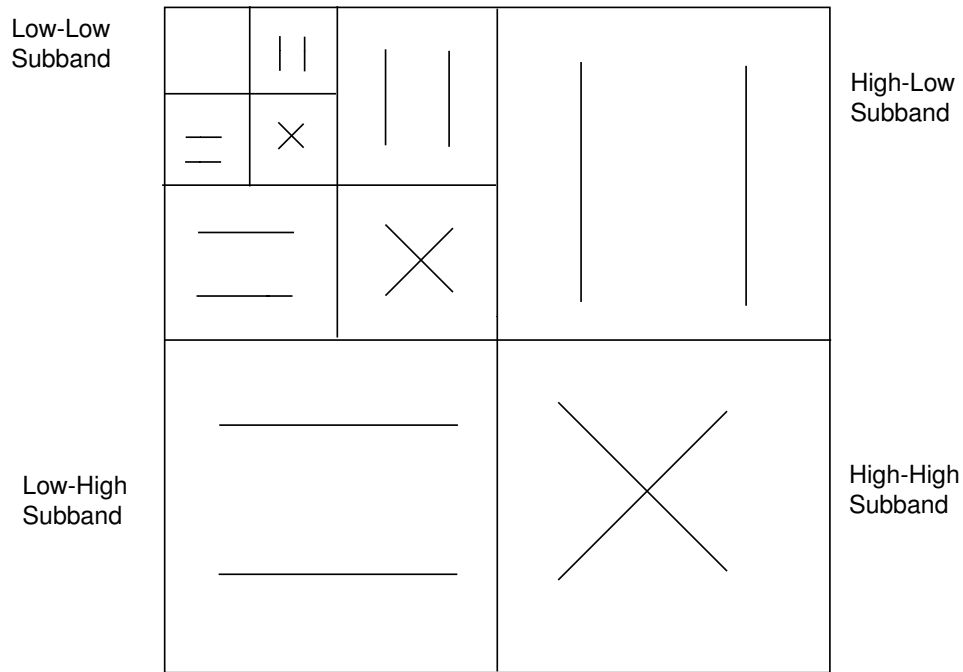


Figure 2.3. *Three Iterations of the Discrete Wavelet Transforms.*

the size of the original image [5]. The RDWT does have some limitations; it is not scale invariant, and it is slower than the discrete wavelet transform. However, for the purpose of our algorithm, these limitations are irrelevant (we are not analyzing scale decompositions) or negligible (we only use three iterations of the transform).

Figure 2.4 shows the implementation of both the discrete wavelet transform (a) and the redundant discrete wavelet transform (b). Notice that the discrete wavelet transform discards information through downsampling, which is why it is not shift invariant. If we were to take the discrete wavelet transform from a different starting point, we would gather different data; this is not the case with the redundant discrete wavelet transform, since there is no downsampling. The structures of the two transforms are similar, but it is clearly seen in the redundant discrete wavelet transform that the number of wavelet coefficients increases with each iteration.

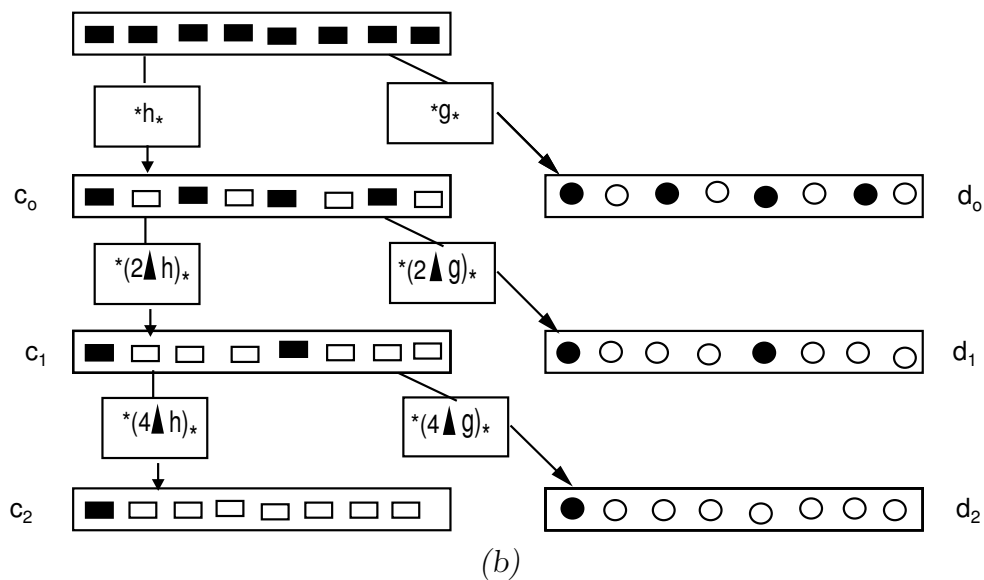
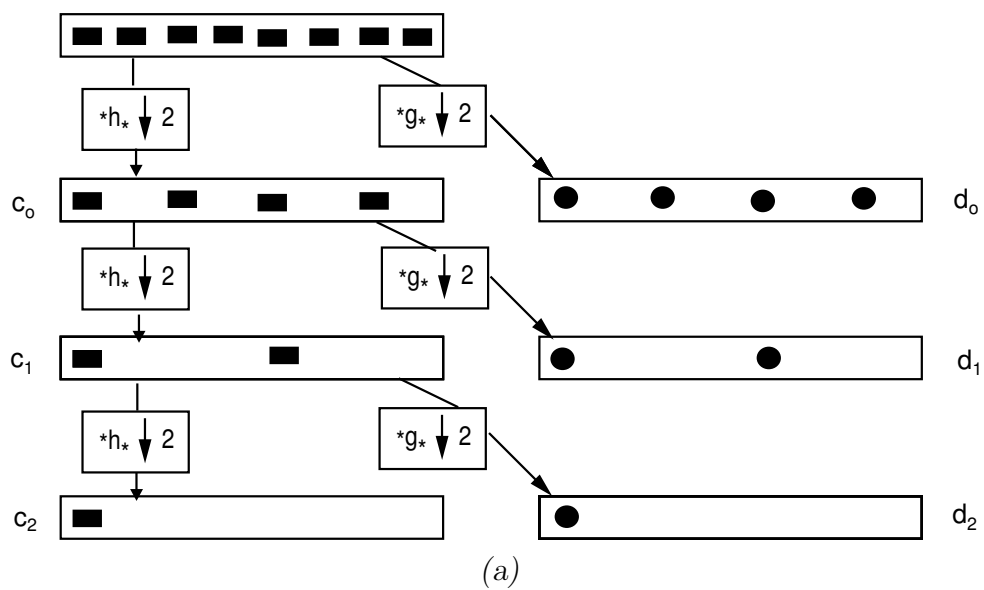


Figure 2.4. *The Discrete Wavelet Transforms. (a) Implementation of the Discrete Wavelet Transform. (b) Implementation of the Redundant Discrete Wavelet Transform which increases in size.*

2.5 Morphology

Morphology, more specifically morphological image processing, is the classification of pixel values in an image using shapes. The shape or the structural element is used to define how the pixels are grouped. For example, our algorithm decomposes and binary thresholds the image. We are left with only '1's and '0's. We wish to find groups of '1' values and combine them into objects. We are able to do this by finding the connectivity of the '1' values. That is, if a '1' is surrounded by a specified number of '1's, they are grouped together into an structural element. A shape is then applied to these structural elements to form an object. The background ('0's) can be defined in a similar process. Certain characteristics can then be determined for each object. These characteristics are, but are not limited to, centroids, bounding boxes and areas. The centroid is the x and y coordinate that defines the center of an object, while the bounding box defines the smallest rectangular region that encompasses the feature. Area defines the actual number of pixels in the feature.

2.6 Optic Flow Overview

Optic flow is defined as the observation of luminance of objects over time. Clearly, image registration can be considered a subtask of optic flow calculations. In general, optic flow is used to find the motion or velocity of objects. For the purpose of our algorithm, we assume that objects in the scene are stationary and that the sensor is moving. Optic flow is currently used to determine motion in many applications, such as computer vision. One of its main disadvantages is that it suffers from the ambiguous constraint that the image must be sufficiently textured. That is, without sufficient texture, invalid estimates are calculated. Unfortunately, no standard definition of texture exists. One of the main contributions of our algorithm is that it defines texture within the content of feature extraction.

Referring to Figure 2.5, point $P(t)$ is our initial sensor starting point. This point moves to $P(t+dt)$ at time dt , and the corresponding velocity vector is $V=dP/dt$.

The projection of our points onto the corresponding image plane are $p(t)$ and $p(t+dt)$. Thus, the apparent x and y velocities in the image plane are $u=dx/dt$ and $v=dy/dt$, respectively. The values $u(x,y)$ and $v(x,y)$ define the motion field or optic flow [14], as shown in Figure 2.6.

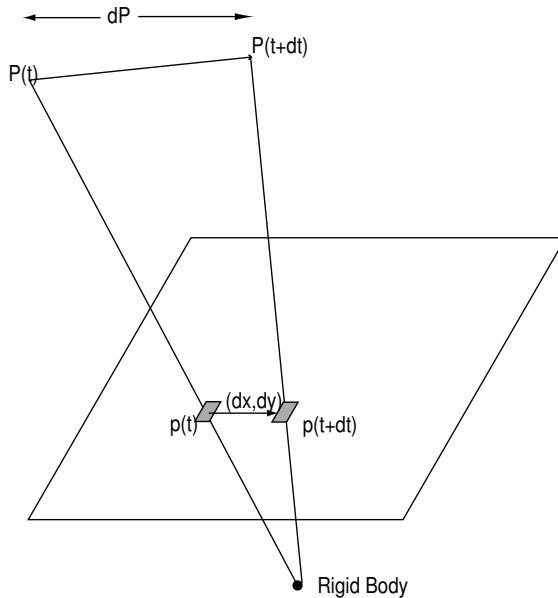


Figure 2.5. *Graphical view of the Motion Field.*

Different techniques are used to estimate u and v at the x,y points. Some general assumptions are stated at this point. It is assumed that illumination (I) changes little with respect to time and that u and v are relatively small, where I is defined, as seen in Figure 2.6, with respect to time as

$$I(x + u(x, y), y + v(x, y), t). \quad (2.1)$$

Thus we have

$$I(x + u(x, y), y + v(x, y), t) \approx I(x, y, t - 1), \quad (2.2)$$

which can be further approximated by

$$I(x + u(x, y), y + v(x, y), t) \approx I(x, y, t) + u(x, y)dI/dt + v(x, y)dI/dy. \quad (2.3)$$

Combining Equation 2.2 and Equation 2.3, we have

$$I(x, y, t - 1) = I(x, y, t) + u(x, y)dI/dt + v(x, y)dI/dy, \quad (2.4)$$

which is the fundamental equation of motion. It can be furthered simplified, and it is more commonly written as

$$I(x, y, t) - I(x, y, t - 1) + u(x, y)dI/dt + v(x, y)dI/dy = 0. \quad (2.5)$$

While this equation gives us a constraint per pixel, u and v are two unknowns that cannot be recovered from the equation unambiguously. Thus, numerous approaches have been developed to estimate these values.

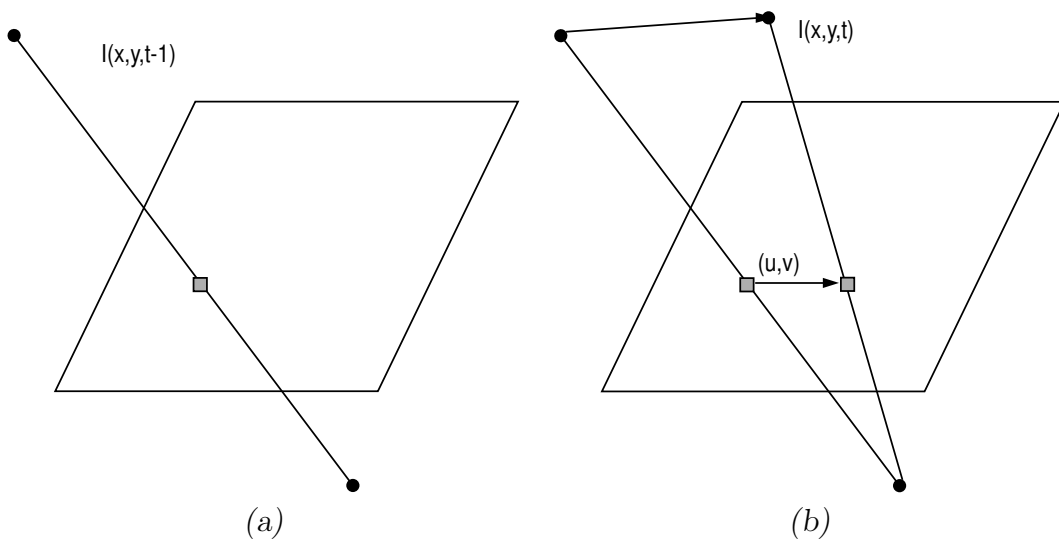


Figure 2.6. *Image plane with respect to time. (a) Motion Field at time = $t-1$. (b) Motion Field at time = t .*

2.7 *Optic Flow Techniques*

Several optic flow techniques have been developed [2, 8, 9, 13, 21, 22, 25, 27, 28]. Most optic flow algorithms focus on either efficiency or accuracy. Both are important for real world applications, yet, previous algorithms struggle to find a happy medium between them [12]. If accuracy is the goal, then the algorithm disregards the need for efficiency (and vice versa). Today, optic flow is moving towards a more accurate and computationally effective algorithm. In a recent review by Barron et al., nine different techniques were evaluated [3]. These techniques fell into the categories of differential, matching, energy-based, and phase-based. The phase-based algorithm proved to be among the more accurate [3]. Fleet and Jepson have demonstrated that phase contours are more stable with small translations in the images, because the phase contours are more robust with smooth shading and light variations [3, 7]. Therefore, Phase-Based Optic Flow is one of the reference technique in this research.

2.7.1 Phase-Based Optic Flow Technique. The Phase-Based Optic Flow Technique used here is based largely on Gautama and Van Hulle [10]. Gautama and Van Hulle take a similar approach to that of Fleet and Jepson [7], except that they spatially filter the input data using a bank of quadrature pair Gabor filters instead of spatiotemporal filtering. Also, Fleet and Jepson use instability as criteria for determining valid phases, whereas Gautama and Van Hulle use non-linearity as their criteria, assuming that phase estimates that are unstable will also be non-linear. Gautama and Van Hulle break their algorithm down into three stages: compute phase response, examine the reliability of components, and calculate velocity components.

1. Compute Phase Response

The phase response is computed for the image sequence by spatially filtering each image with a set of quadrature Gabor filter pairs at every frame. The tem-

poral phase gradient for every filter at every spatial location is also computed. The component velocity is then derived from the temporal phase gradient.

Gabor filters have the benefits of Fourier analysis and also give locality information (like wavelets). Gabor filters are harmonic functions which are similar to Fourier basis functions, but they are limited to certain spatial locations due to their characteristic Gaussian damping terms. This allows them to capture spatial information with the real part of the equation, while the complex part captures frequency information. The two-dimensional Gabor function is represented in polar coordinates by

$$G(\rho, \theta) = \exp(-i\omega(\theta - \theta_o)) \exp\left(\frac{(\rho - \rho_o)^2}{\sigma_\rho^2}\right) \exp\left(\frac{-(\theta - \theta_o)^2}{\sigma_\theta^2}\right), \quad (2.6)$$

where

$$\sigma = \frac{2^\beta + 1}{(2^\beta - 1)2\pi\sqrt{f_x^2 + f_y^2}}. \quad (2.7)$$

Here β is the number of octaves for the constant bandwidth and f_x and f_y are the center frequencies.

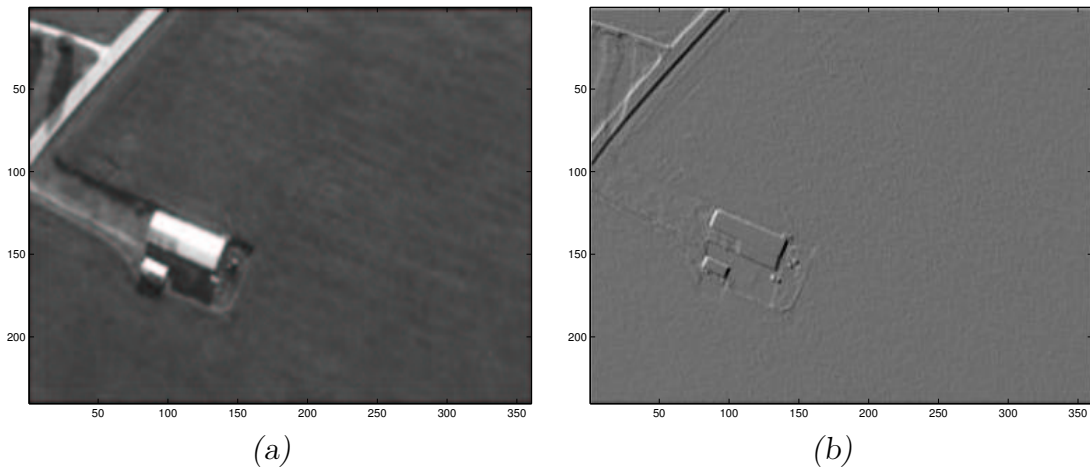


Figure 2.7. *Gabor Filter designs. (a) Real part of the image. (b) Imaginary part of the image.*

As shown in Figure 2.7 and Equation 2.6, the output of the quadrature Gabor filter pair has both real and imaginary parts, where the imaginary part contains the phase information. The temporal phase gradient, which was found using Fleet and Jepson's [7] assumption, provides the component velocities.

2. Reliability of Components

The reliability of components is estimated using phase non-linearity. Previous work by Fleet and Jepson [7] focused on the instability of phase information. The assumption here is that areas of instability are not likely to have phase linearity. To measure non-linearity, a linear least-squares regression is performed on the computed phase, then the mean square error is divided by the absolute value of the estimated gradient as a metric to ensure linearity.

2.7.2 Wavelet-Based Optic Flow. From the basic definition of optic flow, there are many ways, besides phase-based calculations, that displacements can be computed. We chose to apply an additional wavelet-based algorithm to our methodology in Chapter Three. The characteristics discussed in Section 2.4 suggest that wavelets are ideal for this situation. Although wavelets have not traditionally been considered for optic flow, there are potential benefits in attacking the optic flow problem from a different perspective. We use Brown's Wavelet-Based Translation algorithm [5] to gather our estimates for pixel displacements, and then we use the displacement values to calculate our velocities.

2.8 Calculate the Full Velocity

Once the image displacements and component velocities are calculated by further Phase- or Wavelet-Based analysis, the full sensor velocity must be calculated. Component velocities are naturally noisy and are given a constraint line for the best fit.

2.9 *Summary*

This chapter documented the reason that wavelets are a natural choice for image registration problems. The basic properties inherent to wavelets are necessary for extracting proper features for our algorithm. A general background on wavelets, the discrete wavelet transform, and the redundant discrete wavelet transform are given to provide a deeper understanding of our methodology, presented in Chapter Three. This chapter also includes morphology and optic flow fundamentals. We also discuss our reasoning for choosing both Phase- and Wavelet-Based Optic Flow algorithms to test the accuracy, efficiency, and robustness of our algorithm. Overall, this chapter documents the basic theory that we exploit in the design and development of our Feature Guided algorithm.

III. Methodology

3.1 Introduction

We require that our Feature Guided Image Registration algorithm be automatic, general, robust, efficient, and accurate. First, the algorithm must be able to function with little human interaction (automatic). Second, it must be general enough to handle images of different topographical data. Third, it must be robust enough to handle translated images in the presence of noise. Fourth, it is important that our algorithm be computationally efficient, since the long term goal is use in a ‘real-time’ navigation system. Finally, it is important that the algorithm be accurate, since it will be used to aid in navigation. To meet these needs we propose a Wavelet-Based Feature Extraction algorithm.

Wavelet transforms are natural edge detectors and can be used to accurately detect features. We discussed why wavelet transforms are useful in image registration problems in Section 2.4. Our proposed algorithm uses the redundant discrete wavelet transform (RDWT) to detect persistent wavelet maxima. It then uses morphological imaging to group maxima into objects. The objects are evaluated so that each has the characteristics of area, centroid, and bounding box. These characteristics are used to define regions of interest (ROI). The initial image is decomposed into subimages based on the ROIs. The same ROIs are imposed on the next image in the sequence, so that the same area is covered in the subimages. These subimages may be processed in parallel to decrease computational time. The corresponding subimages are then used in a Phase-Based Optic Flow algorithm from Gautama and Van Hulle [10] and a Wavelet-Based Translation algorithm from Brown [5] to calculate pixel shifts for synthetic data and velocities from a real video stream. The testing and overall assessment of these algorithms is presented in Chapter Four.

3.2 Wavelet-Based Feature Extraction

The purpose of Wavelet-Based Feature Extraction is to highlight the image areas that have dominant features and then calculate the pixel shift in that area. By calculating estimates of the pixel shift in multiple regions, we average the estimates and improve accuracy. We also determine the amount of texture in a image by the number of features we extract. Our algorithm states when insufficient persistent features are present, which enables it to indicate that an unreliable pixel shift may have been estimated.

It is critical, for accurate velocity estimation, that proper features are extracted by our algorithm. The criteria of Li and Zhou [20] are used. To ensure that our algorithm is robust, spacing is required around the feature. If there is not enough area around the feature it may not be present in the sequential image and an invalid estimate is calculated. The bounding boxes must be large enough to ensure that accurate translation estimates are computed and yet small enough to minimize computational complexity. Thus, each bounding box is proportional to the feature, making the box size adaptive. The same bounding boxes are compared in the two sequential images to ensure consistency.

The features extracted are based on edges, which are naturally regions of high contrast. Since edges are usually spread throughout the image, we assume that the clustering of wavelet coefficients is distributed similarly. To meet Li and Zhou's [20] final criteria, the features must be unique, which is possible, since the wavelet coefficients are naturally parsimonious, and we threshold the coefficients and reduce background noise levels. This procedure implies that the same unique feature is extracted even in the presence of noise.

3.2.1 Pre-Filtering the Data Imagery. Since one of the main goals of our algorithm is to increase computational efficiency of the Phase-Based Optic Flow and Wavelet-Based Translation Algorithms, we apply a high quality compression filter

to the video stream data. This filter reduces the size of each frame by a two to one ratio in both directions by lowpass filtering and downsampling, which decreases the size of our original image from 480×720 to 240×360 as shown in Figure 3.1. The images are also converted into grayscale. Our synthetic data, which is used for validation of our algorithm, does not need to be filtered, because it is already a grayscale 256×256 noise-free image.

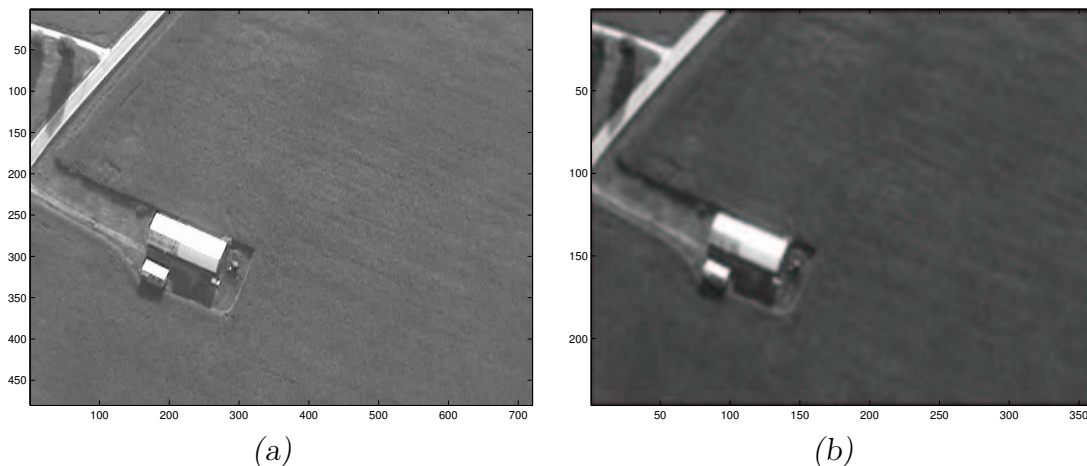


Figure 3.1. *Original Images. (a) Original Image, which also has a third color dimension which is not shown here. (b) Original Image after filtering and size reduction.*

3.2.2 Selection of Wavelets Basis for Feature Extraction. To validate the robustness of our algorithm, we analyze various wavelet decompositions to determine which cause more features to be extracted. For these tests we use the “Lenna” image. Table 3.1 shows some of the 45 Daubechies wavelets and the 15 biorthogonal wavelets used in our analysis. The ‘db2’ wavelet extracts the largest number of features and produces low computational complexity. Thus, it is the choice for our feature based algorithm.

3.2.3 Shift-Invariant Wavelet Transform. Since it is critical that we extract the same features at every iteration and maintain feature consistency from frame to frame, the shift-invariant wavelet transform is the obvious choice for decomposing

Table 3.1. *Number of features per wavelet decomposition.*

Wavelet Name	Number of Features	Number of Features with Noise
db1	8	3
db2	23	2
db3	18	2
db4	22	2
db5	22	2
db6	21	2
db7	21	2
db8	21	2
db9	19	2
db10	20	2
db15	15	2
db20	16	2
db25	12	3
db30	10	2
db35	9	3
db40	10	2
db45	18	2
bior1.1	8	2
bior1.5	8	2
bior2.8	17	2
bior3.9	20	2
bior4.4	20	2
bior5.5	20	2
bior6.8	20	2

Table 3.2. *Number of features per RDWT iteration.*

Number of Iterations	Number of Features
2	23
3	23
4	23

our image. The redundant discrete wavelet transform we use provides the shift invariant properties we desire. Figure 3.2 shows the filter bank representation of the RDWT. Notice that there is no down-sampling, which ensures that each subband of the image stays the same size.

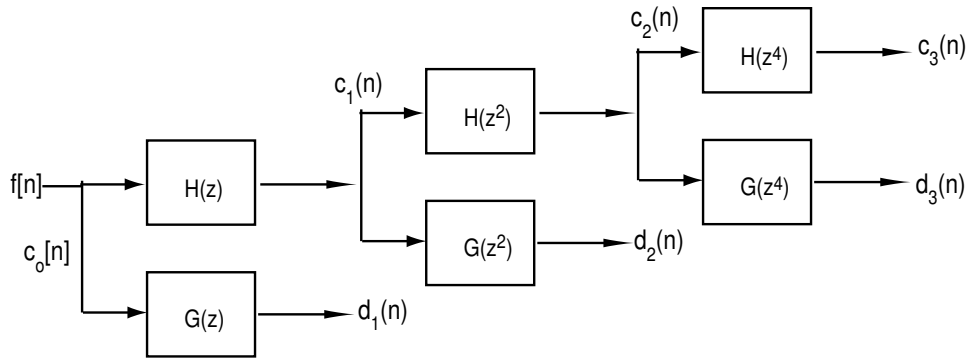


Figure 3.2. *Filter bank representation of the Redundant Discrete Wavelet Transform.*

It is important that we compute the redundant discrete wavelet for a sufficient number of iterations to determine that a feature is dominant. Table 3.2 shows that the use of two, three, and four iterations yields 23 features for the Lenna image. Three is chosen because it extracts the same number of features as four iterations without extra computation. Two was not chosen, because two iterations did not seem to guarantee that the features would be present in the future iterations.

3.2.4 Subband Choices. The HL and LH subbands, described in Section 2.4, are chosen for analysis because they are the most useful for image registration. The HH subband is not used, because it preserves high frequency components and is therefore the most susceptible to noise, which is an undesirable trait. The LL

subband is undesirable because the transform coefficients are not sparse. The LH subband tends to preserve the horizontal features, while the HL preserves the vertical features. These subbands prove useful for calculating the pixel shifts and velocities.

3.2.5 Lowpass Filter. The next step in our process is to apply a lowpass filter to the LH and HL subbands. The purpose of the lowpass filter is to smooth the edges of the image. It also allows for the correction of the slight shift in the images due to the odd length wavelet filter that may be chosen. When an odd length wavelet filter is chosen, the RDWT shifts one pixel due to padding in the circular convolving process. This shift proves to be problematic in the next section of our algorithm. If subsequent iterations of the image are shifted, it is difficult to combine the subbands and extract features. The lowpass filter also helps reduce the noise that may still be present in the image, which improves the robustness of our algorithm, since we hope to extract the proper features regardless of noise level.

3.2.6 Threshold and Binary Masking. Once the noise has been reduced and the redundant discrete wavelet transform shifts accounted for, we hope to reduce the complexity of our wavelet representation. Since wavelets are known for being parsimonious, it is unnecessary to keep all coefficients. We apply a threshold to the coefficients, and only retain those coefficients whose magnitude exceeds the threshold. We use the Guo, Odegard, et al., threshold of 3σ [11], where σ is the standard deviation given by

$$\sigma = \sqrt{\frac{1}{n} \sum ((x_i - \bar{x})^2)}, \quad (3.1)$$

where

$$\bar{x} = \frac{1}{n} \sum (x_i). \quad (3.2)$$

The x_i are the wavelet coefficients, and n is equal to the number of coefficients. This threshold minimizes the effects of noise, since it is likely that the significant

coefficients exceed the threshold even with noise added. Since the 3σ value may be different for each iteration of the RDWT, there may be discrepancies in the number of significant coefficients retained. To solve this problem, we use Guo and Odegard’s threshold [11] on the first subband and determine the number of significant coefficients. We then adjust the threshold at subsequent scales to ensure that a consistent number of coefficients remain throughout the iterations.

The significant coefficients are then set to ‘1’ and all others are set to ‘0’. This binary masking is very computationally efficient and further reduces the effects of noise. For our purposes we are only concerned with the existence of a feature and not its magnitude. The subbands of the different RDWT iterations are then logically *anded* together. If a ‘1’ is present in all iterations of the RDWT then it is assumed that it is a dominant feature, due to persistence.

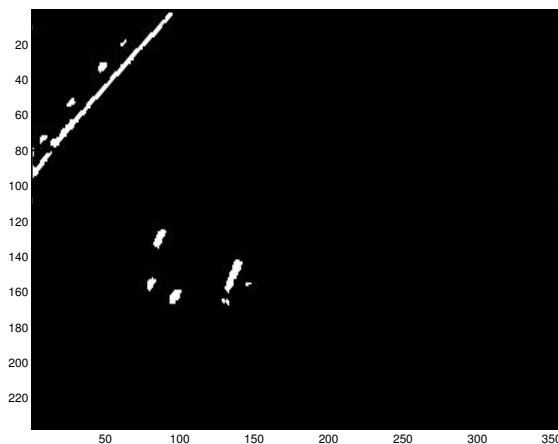


Figure 3.3. *The anded HL subband after the threshold is applied and the significant coefficients are set to ‘1’.*

3.2.7 Morphological Imaging. At this point, we have binary subbands with ‘1’s at each of the feature pixels. We now use morphological processing to group these pixels into true features. Our morphological imaging takes place in four steps:

1. The first step in our morphological processing is to open the image with a structural element. For all of our test data we used *shape = disk* and *radius =*

15. This process removes any independent ‘1’ that cannot be contained in our structural object as shown in Figure 3.4.

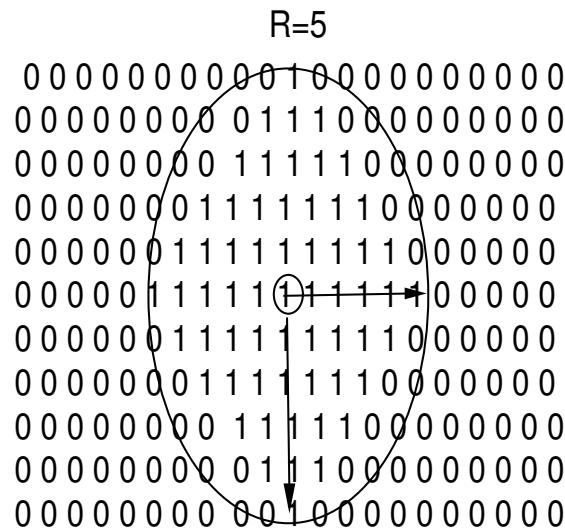


Figure 3.4. Size and the shape required to make a group of ones into an object. For this figure the radius is 5; for our algorithm it is 15.

2. The second step is to find an approximation of the background. This is done by using the MatLab Version 12.1 *surf* command. The *surf* command provides the characteristics of the background, which enables us to eliminate independent ‘1’s that may be present due to noise or other distortions.
3. The third step is to find all connecting ‘1’s and label them. Different labelling techniques are used for the different subbands. The HL subband uses a two dimensional labelling technique, *bwlabel*, since the algorithm provided by MatLab Version 12.1 is designed to calculate two-dimensional objects in the vertical direction most efficiently. However, the multidimensional technique, *bwlabeln*, is faster calculating horizontal two-dimensional objects. Therefore, this command is used in the LH subband. This process defines the connectivity that an object must have to be considered an object.
4. Finally, properties of the objects are formed. Once all the objects are labelled and identified, it is possible to find centroids and bounding boxes. The centroid

is the x and y coordinate of the center point of the feature. The bounding box defines the smallest rectangular region that encompasses the feature.

If no objects were created during the morphological process, our algorithm tells the user that zero features were extracted, and the image is considered ‘bad’. Our algorithm then proceeds to skip the following steps and goes straight into the Phase-Based Optic Flow algorithm or the Wavelet-Based Optic Flow algorithm. The ability to determine the absence of features or texture is a key contribution, and allows us to determine when estimates are not meaningful.

3.2.8 Creating Regions of Interest (ROI). By creating objects out of the groupings of ‘1’s, we are able to find the centroid and the bounding box of each feature. We take the dimensions of the bounding box and ensure that they are at least 60×90 , depending on image size. This minimum size is determined experimentally with the Phase-Based Optic Flow algorithm. The image is then cropped as defined by the bounding box. The minimum size ensures that we detect a considerable translation, while cropping the bounding boxes ensures that we gain computational efficiency from creating subimages. Figure 3.5 shows three significant features extracted using our algorithm on frame zero of the video stream data.

Once the ROIs are determined from the reference image, the same areas are selected on the sequential image, which allows our algorithm to be feature guided. We extracted the features from the reference image and then found ROIs. These ROIs are then imposed on the sequential image so that we can crop the image. The separate subimages can then be used in the optic flow algorithms. This Feature-Based algorithm provides for increased computational speed, since the images are significantly smaller and can be processed in parallel. Figure 3.6 shows the ROI of feature two (as defined in Figure 3.5 (c)) in the original image (a) and the sequential image (b).

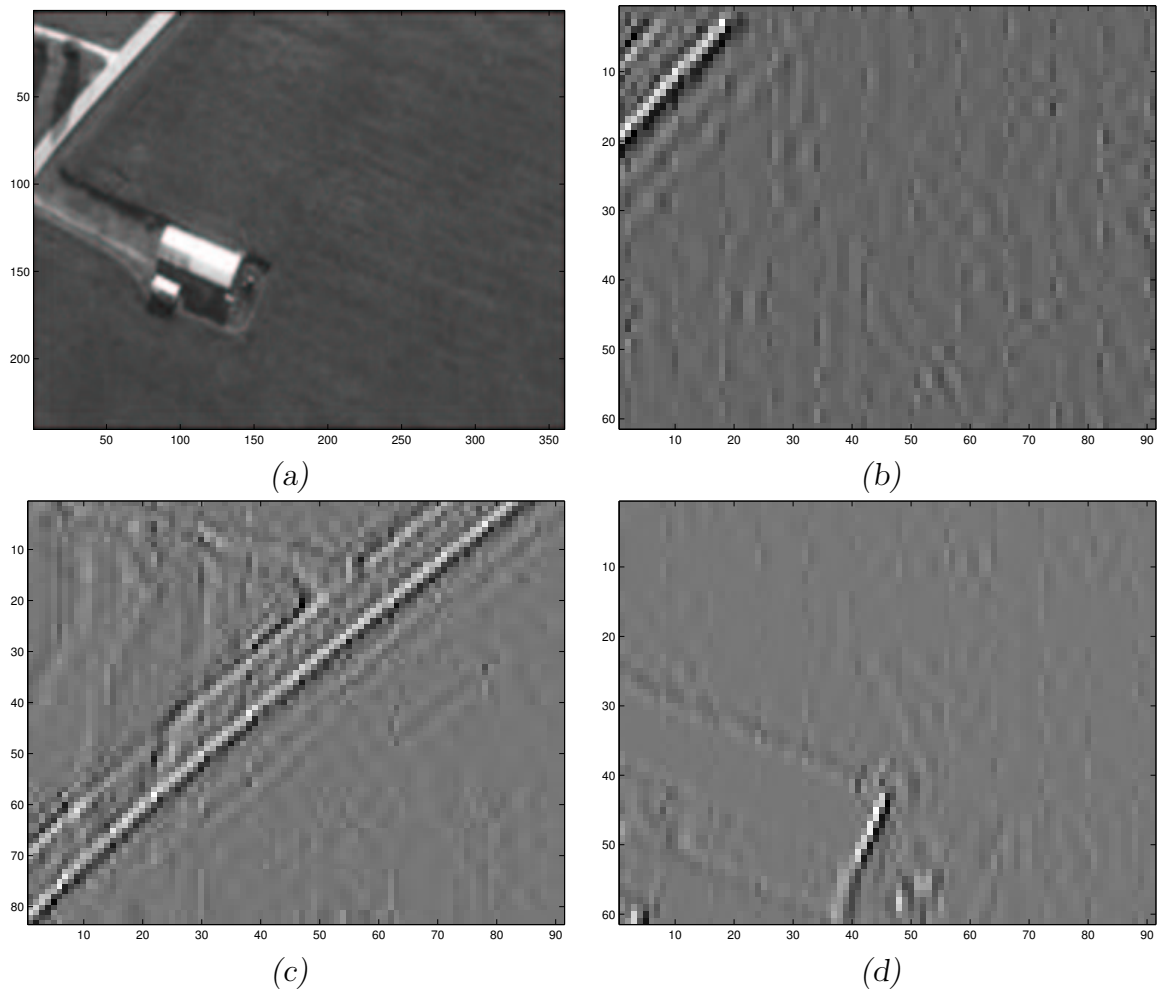


Figure 3.5. *Subimages cropped from the original image. (a) Original Image. (b) First Feature. (c) Second Feature. (d) Third Feature.*

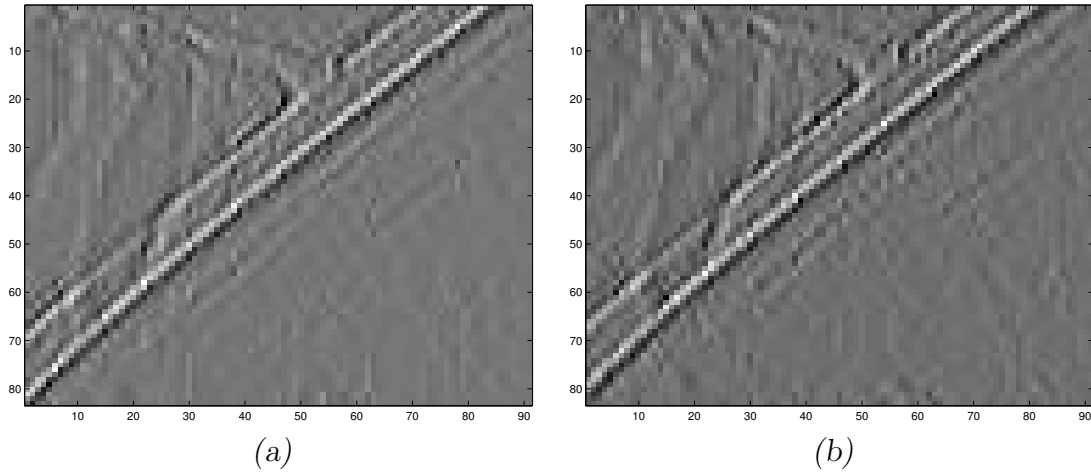


Figure 3.6. *Two Regions of Interest to be compared. (a) Original ROI of feature two on the reference image. (b) Original ROI of feature two imposed on the input image.*

3.2.9 Additional Parameters that Modify our Code. To ensure that proper translation is calculated, we make sure that proper feature extraction takes place. Two additional parameters are taken into account when designing our algorithm. They are determining what is a ‘good’ and ‘bad’ image. A good image is one that has one or more extractable features with adequate area, and a ‘bad’ image has none. Table 3.3 show the translation estimates along with the number of features and their area. It can clearly be seen that area is an important criteria for extracting ‘good’ features for estimating translation.

Table 3.3 is created using the Feature Guided Brown algorithm [5]. From this table, we see that the area helps determine if we have a valid feature, and thus we modify our code accordingly. Logical arguments are implemented as follows: we first determine if more than five features are present. If more than five features are present we implement area constraints. We test to see if we have at least two features of area 25 pixels or greater. If this is true, then we precede with our algorithm as stated above. If this is false, we determine if we have at least two features of area 10 pixels or greater. If this argument is true, we then precede with the rest of our algorithm. If these criteria are false, then we use all the features extracted regardless

Table 3.3. *Number of Features Extracted along with their corresponding areas and translation estimates in the x and y direction.*

Feature Numbers	Area	Tx	Ty
1	49	2	0
2	43	2	2
3	30	3	0
4	18	2	2
5	17	2	2
6	17	2	2
7	13	2	2
8	12	2	2
9	11	2	2
10	10	2	2
11	5	2	2
12	4	2	2
13	4	2	2
14	3	2	2
15	3	2	2
16	2	2	2
17	1	2	2
18	1	2	2
19	1	2	2
20	1	2	2
21	1	2	1
22	1	1	0
23	1	75	0

of area. This procedure allows the algorithm to make an estimate in all cases, but it places emphasis on the most significant features.

The next modification of our algorithm is imposed by the constraints of the Phase-Based Optic Flow algorithm. This algorithm uses a quadrature Gabor filter bank, which produces edge effects of approximately 20 pixels and can produce ‘ugly’ features. ‘Ugly’ features are defined as features contained mostly in the 20 pixel outer region corrupted by the Gabor filter. To compensate for this problem we add an exclusion region around the original image, which is done by imposing ‘0’s around the border in the binary image. Figure 3.7 shows examples of an ‘ugly’ and a ‘bad’ image.

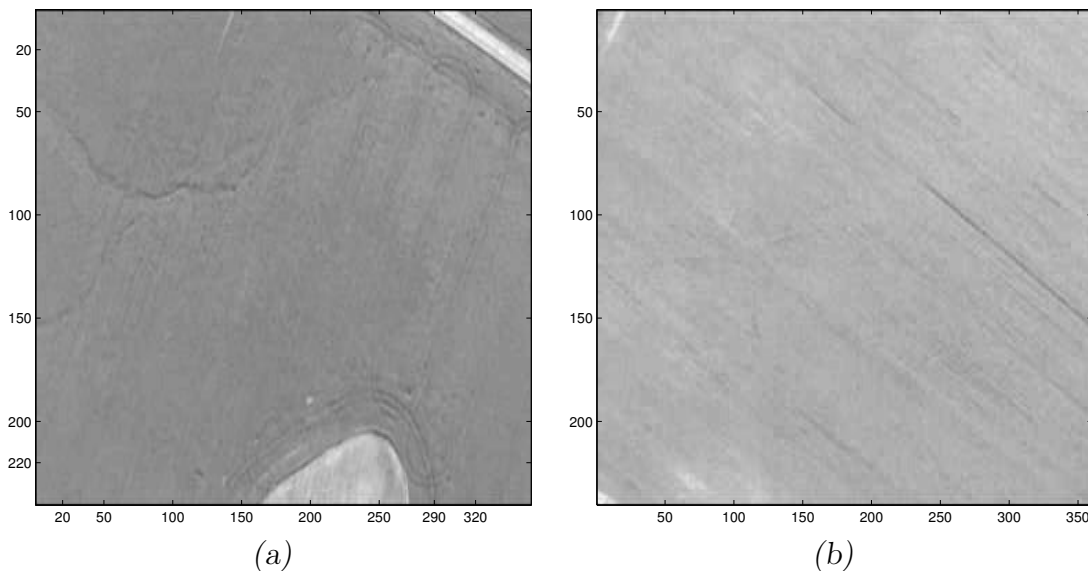


Figure 3.7. *Two frames of the video stream data that show the ugly and the bad features. (a) ‘Ugly’ feature for frame 850 of the video stream data. Features are only located near the edges where the edge effects due to the Gabor Filter produce invalid estimates. (b) ‘Bad’ or no feature image from frame 584 of the video stream data. No features are present.*

3.3 Wavelet-Based Translation Algorithm

The Wavelet-Based Translation algorithm that we use to validate our Feature Guided algorithm is based heavily on Brown [5]. With our feature extraction methodology we create subimages from the original image based on features being present, which allows us to reduce Brown's [5] computational complexity by

1. The ability to process the images in parallel.
2. The fact that a smaller number of coefficient can be processed to yield the same accuracy.

The following describes the basic methodology of Brown's [5] algorithm and how it is used in our Feature Guided algorithm.

3.3.1 Number of Significant Coefficients. The number of significant coefficients (N) is defined by the user. It usually varies from 10 to 250 [5]. These are the coefficients that are used to form our translation estimate. Computational complexity increases as N^2 ; since our subimages are small (typically 60×90), the speed up of Brown's [5] algorithm is significant.

3.3.2 RDWT. One iteration of the RDWT is performed on the images. One iteration is considered adequate in this situation, because each subband is the same as the original size. There is no gain in computational efficiency, since each further iteration is the same size as the original image.

3.3.3 Masking. Masking for this algorithm is done differently than in our algorithm: N is determined by how many significant coefficients are chosen, and not by 3σ as in our algorithm. This enables the Brown algorithm [5] to be more flexible but does not offer the automation of our algorithm.

3.3.4 Comparison of Location. The location of the most significant coefficient in the reference subimage is compared to all the N significant coefficients in

the input subimage, which produces N translation estimates. This procedure is then repeated for the middle and the 10th largest significant coefficient, which gives us $3N$ translation estimates per subband. As mentioned in Section 3.3.2, our N is typically smaller.

3.3.5 Correlations. Correlations are performed for the $3N$ estimates calculated above. The coefficients are circularly shifted and compared. The highest correlation is considered the best estimate, which is done with the LH and the HL subbands separately. A Pearson Correlation in the spatial domain is then done with the two highest correlating estimates in each subband [5]. The estimate with the highest Pearson Correlation is the final pixel shift estimate. This estimate is used in the velocity calculation presented later in this chapter.

3.4 Phase-Base Optic Flow Algorithm

As discussed in Section 2.7.1, phase-based optic flow algorithms are considered among the more accurate, but their computational complexity is a disadvantage. Since we have reduced the image sizes considerably and process the subimages in parallel, we are able to compute a faster and more effective result. Our Phase-Based algorithm is based heavily on Gautama and Van Hulle [10] with a few adjustments.

3.4.1 Methodology for Optic Flow Algorithm.

3.4.1.1 Linearity Threshold. The linearity threshold is very critical to the second step of Gautama and Van Hulle's algorithm [10]. It determines criteria for the rejection of non-linear phase components. It is extremely important that the phase be linear, since most spatial information is located in the phase. If the phase becomes distorted (i.e., non-linear) our estimates are corrupted. As this constraint tightens, the deviation between component velocities decreases (desirable) and the density coverage of the optic flow field also decreases (undesirable). The coverage

defines the percentage of the image which is able to give valid velocity estimates. For our experiment, a linearity threshold of .01 is used. This threshold is used in Gautama and Van Hulle’s algorithm [10], and it is logical to keep it consistent so that we make valid comparisons.

3.4.1.2 Optic Flow. The optic flow is computed with the methodology explained in Section 2.6. An additional constraint is placed on the optic flow calculation to reduce edge effects. An offset is calculated to match where the Gaussian envelope drops below 10 percent, which is necessary because of the quadrature Gabor filter banks characteristics described in Section 2.7.1. The Gabor filters produce edge effects, and since we know the spatial location of the effects, we are able to compensate. Figure 3.8 shows the calculations of the optic flow vectors of feature two. The offset can be clearly seen.

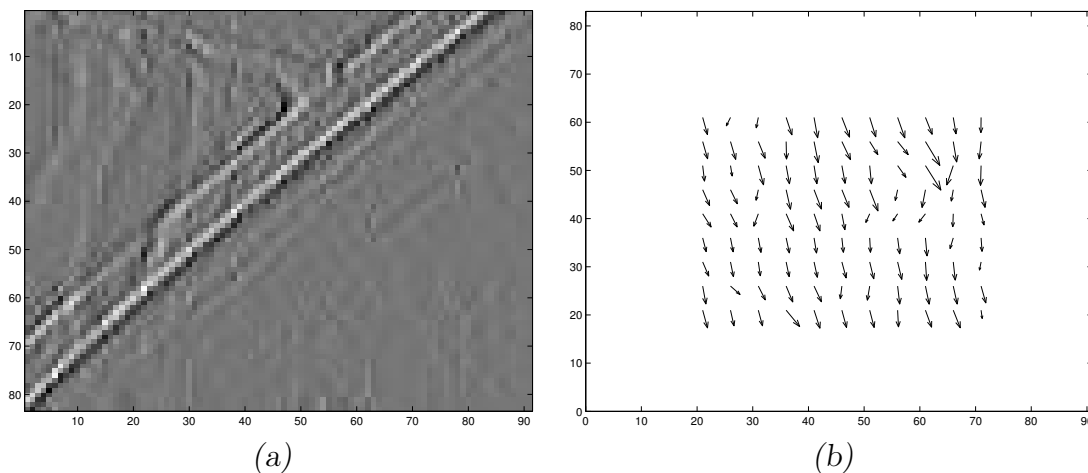


Figure 3.8. *Optic Flow Vectors. (a) Original ROI of feature two on the reference image. (b) Calculated Optic Flow Vectors.*

3.5 Velocity Computation

Once the x and y velocity components from the Phase-Based Optic Flow and the Wavelet-Based Translation algorithm are calculated, this information can be combined with the field of vision (FOV) to calculate the final velocity vector.

The FOV and the pixel resolution are predetermined theoretically and then recalculated with real data to provide an accurate estimate. These parameters are determined by the characteristics of the camera being used.

We chose to sample the video data at 15 frames per second. We are still able to calculate the proper estimates with downsampling. If we were to use the full 30 frames per second of data, we would calculate invalid velocities, since the displacement, due to compression, is too small to notice (less than .1 pixel per second).

3.5.1 Methodology for Altitude Calculation. Our video stream data is gathered using the Global Positioning system (GPS). Our data file contains altitude in terms of Mean Sea Level (MSL). For our algorithm we are interested in the object-to-camera altitude. Thus we must convert the MSL altitude to terrain-to-platform altitude, which can be done by taking the MSL altitude and subtracting the height of the terrain, with respect to mean sea level.

3.5.2 Velocity calculation. The velocity calculation is based on the parameters defined above. The actual calculation is

$$f_{xp} = altitude \times \tan\left(\frac{FOV_x}{p_x}\right) \quad (3.3)$$

and

$$f_{yp} = altitude \times \tan\left(\frac{FOV_y}{p_y}\right), \quad (3.4)$$

where f_p is the displacement in the field of motion in the respective direction and p is the number of pixels in the respective directions. The velocity is then given by

$$velocity = \sqrt{(f_{xp} \times V_x)^2 + (f_{yp} \times V_y)^2} \times rate_{frame} \times \frac{3600}{5280}, \quad (3.5)$$

where V are the velocity components. The constant at the end of Equation 3.5 converts the velocity into miles per hour: 3600 is equal to the number of seconds in an hour and 5280 is the number of feet in a mile.

An additional criterion is used to help reduce further errors. We include another parameter called the margin of error, ϕ . Where ϕ is

$$\phi = \arctan\left(\frac{V_x}{V_y}\right), \quad (3.6)$$

This margin is based on finding the largest velocity vector and validating that vector and those within the margin of error. First, we estimate the probability density function of the velocity by computing a histogram. We then find the maximum velocity component. The angle of this maximum velocity component is our base angle. The margin of error is then added and subtracted to this maximum. This calculation is based on the assumption that noise vectors are relatively small compared to the true velocity vectors. These are considered valid angles and the corresponding field of motion calculations are inserted in the velocity of Equation 3.5.

3.6 Summary

In this chapter, we describe our methodology for providing a more computationally effective algorithm for determining optic flow pixel shifts. Our algorithm utilizes the redundant discrete wavelet transform, binary masking, and morphological processing for feature extraction and allows us to break apart the images into subimages so that they can be processed faster and in parallel. We then process the subimages with both Phase- and Wavelet-Based Optic Flow Algorithms.

In the next chapter, we discuss the performance of our Feature Guided Phase- and Wavelet-Based Optic Flow Algorithms and compare them to the original Phase- and Wavelet-Based Optic Flow Algorithms. We also discuss the design trade-offs that are made and why.

IV. Results

4.1 Introduction

The purpose of this chapter is to analyze specified data sets to demonstrate the validity and performance of our algorithm. First, we give a general description of our validation study, followed by an explanation of why certain measures of performance and parameters are used. The results of the Phase-Based Optic Flow algorithm versus the Feature Guided Phase-Based Optic Flow algorithm are then presented. Next, the Wavelet-Based Optic Flow algorithm versus the Feature Guided Wavelet-Based algorithm results are documented, which demonstrates how our algorithm improves accuracy. We then test the robustness of the Feature Guided algorithms with various noise and pixel displacements. Finally, we present an overall assessment of how the Feature Guided algorithms meet the criteria, set forth in Chapter Two, of being robust, automatic, and computationally effective.

4.2 Discussion of Validation Study

The validation study compares our Feature Guided algorithm to the original Optic Flow algorithms by processing test images and varying algorithm parameters. The purpose of our research is to create a robust, automatic, and computationally effective algorithm. The validation study results presented here demonstrate that our algorithm meets these criteria better than the original algorithms. Our validation study compares the original Phase- and Wavelet-Based Algorithms to the Feature Guided Phase- and Wavelet-Based Algorithms. We choose to validate our algorithm's flexibility by applying it to two different optic flow algorithms. We show how our algorithm increases the accuracy and robustness in each case.

4.2.1 Test Images. The validation images of our algorithms consist of a synthetic test image "Lenna" and a 60 second video stream of real data. The

synthetic image is shown in Figure 4.1 (a); it is a 256×256 pixel 8-bit grayscale image. The “Lenna” image is used in Brown’s research [5], thus giving us a logical test choice for making performance comparisons. The video stream data is captured from a JAVA miniDV camcorder mounted to a Zenair CH-701 aircraft flying north of Dayton, Ohio at approximately 1,000 feet. The video stream data are originally color, but are converted into a 240×360 8-bit pixel grayscale image before processing by our algorithm. Figure 4.1(b) shows frame zero of the video stream data.

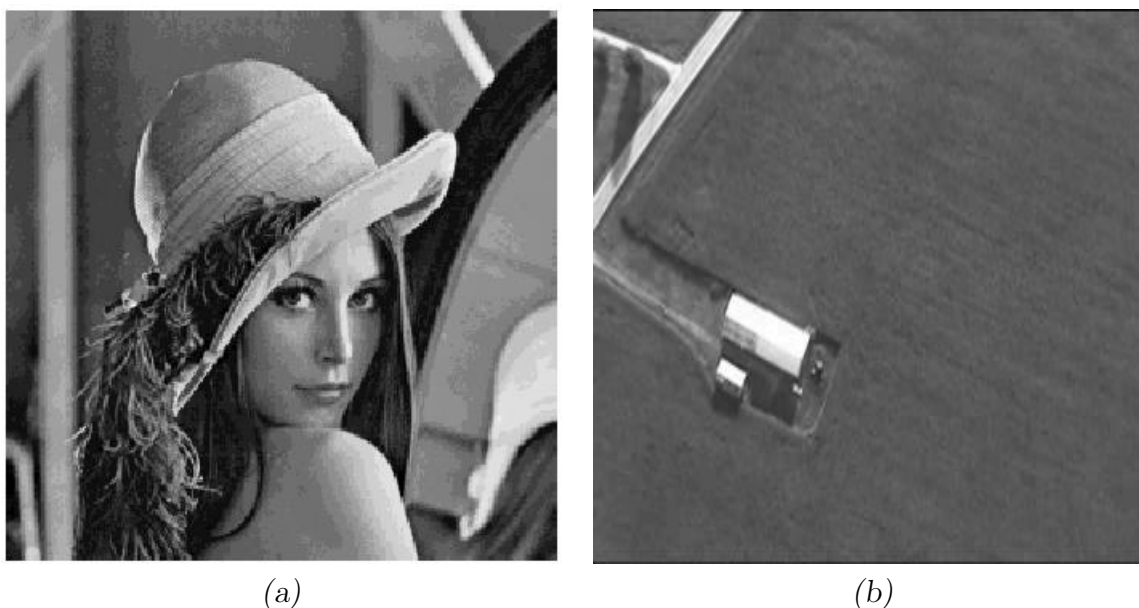


Figure 4.1. *Test Images. (a) “Lenna” synthetic image. (b) First frame of the video stream data.*

4.2.2 Peak Signal-to-Noise Ratio. To validate our algorithm with the synthetic data, it is necessary to add white Gaussian noise to simulate real conditions. We use Peak Signal-to-Noise Ratio (PSNR) to define how the signal is degraded by noise. PSNR is

$$PSNR = 20 \log \left(\frac{\max |x_i|}{\sqrt{\sum_i \frac{(x_i - \hat{x}_i)^2}{N}}} \right), \quad (4.1)$$

where the x_i represent the grayscale values of the pixel in the original image, the \hat{x}_i are the grayscale values of the noisy image, and N is the number of pixels in the original image. For 8-bit images, $\max|x_i| = 255$.

For our validation study, PSNR values of 22 dB and 28 dB are created. Images that have a value of approximately 30 dB or higher are considered to have little noise degradation from the original image. The value of 22 dB is chosen for comparison to the work of Brown, where images with PSNR of 22 dB are studied [5], and 28 dB is chosen because at higher PSNR values the Brown algorithm is documented to have nearly perfect translation estimation performance [5]. In this case our Feature Guided algorithm would only provide faster computation, with no increase in accuracy. Figures 4.2 (a) and (b) show the “Lenna” image with a PSNR of 28 dB and 22 dB respectively. It can be seen that the PSNR is inversely proportional to the level of noise in an image; the higher the PSNR value, the less noise present in the image, as clearly seen in Figure 4.2.

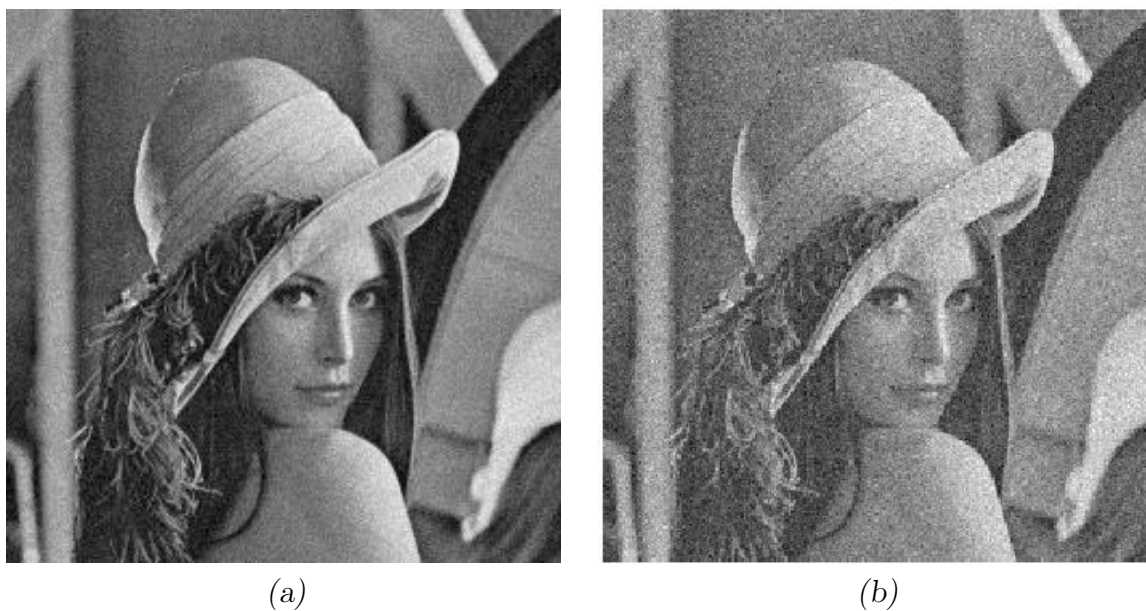


Figure 4.2. Examples of PSNR values for the test images. (a) Lenna image with 28 dB PSNR. (b) Lenna image with 22 dB PSNR.

4.3 Experiment

Our experiment starts by testing the accuracy of our algorithm. We validate our algorithm by testing the original algorithms versus the Feature Guided algorithms. To show robustness with respect to noise, we demonstrate accuracy results with noise of 28 dB and 22 dB. We take feature guided algorithms and compare their accuracy with respect to pixel shifts.

4.3.1 Synthetic Data. A general standard in validating new algorithms uses synthetic data. Since Brown [5] uses the “Lenna” image, it is logical that we do the same. The synthetic data allows us to test for accuracy and robustness with known solutions.

4.3.1.1 Accuracy and Robustness Test with Respect to Noise. To validate the accuracy of our Feature Guided Registration algorithm, we test the original algorithms versus the Feature Guided algorithms with image PSNRs of 28 dB and 22 dB. We show the results for 100 trials. Figure 4.3 (a) and (c) show results for a displacement of two pixels in the x direction. Figures 4.3 (b) and (d) show results for a displacement of two pixels in the y direction. Figure 4.4 shows the same results at 22 dB. Table 4.1 gives additional statistics for the results of Figures 4.3 and 4.4. Table 4.1 shows that the Feature Guided algorithms outperform the original algorithms. While the standard deviation does not change significantly for the Guatama and Van Hulle algorithm, it has a large variation for the Brown algorithm. The graphs in Figure 4.3 show that the Brown algorithm estimates are either extremely accurate or extremely inaccurate, as verified by the standard deviation. It can also be seen in Figure 4.3, Figure 4.4, and Table 4.1 that the Guatama and Van Hulle algorithm is more robust to noise.

4.3.1.2 Accuracy Test with Respect to Pixel Shifts. It is clearly seen from Figure 4.3 and Figure 4.4 that the Feature Guided algorithms outperform

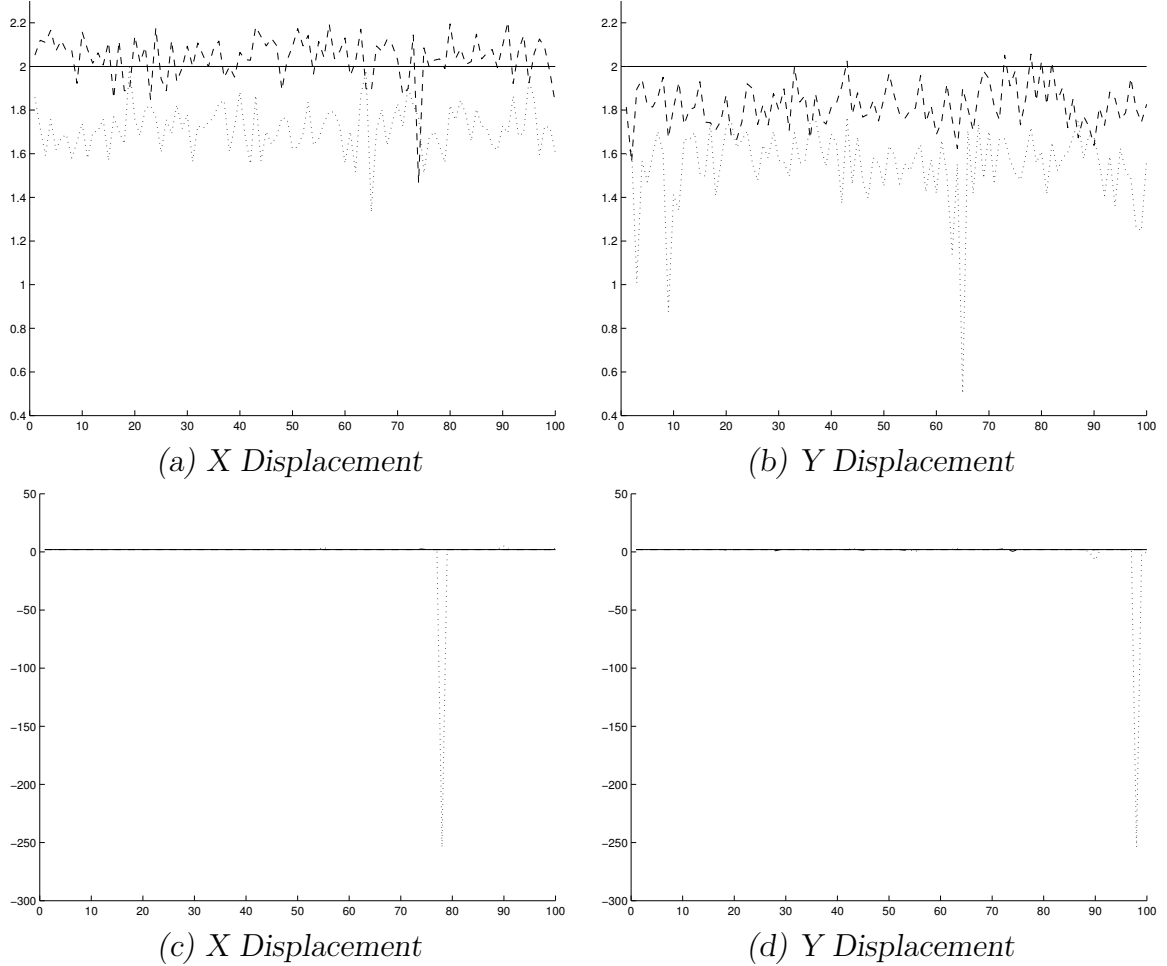


Figure 4.3. Evaluation of Accuracy of pixel shift of two with PSNR of 28. (a) Feature Guided Gautama and Van Hulle algorithm (dashed line) versus Original Gautama and Van Hulle algorithm (dotted line). (b) Feature Guided Gautama and Van Hulle algorithm (dashed line) versus Original Gautama and Van Hulle algorithm (dotted line). (c) Feature Guided Brown algorithm(dashed line) versus Original Brown algorithm(dotted line). (d) Feature Guided Brown algorithm (dashed line) versus Original Brown algorithm (dotted line).

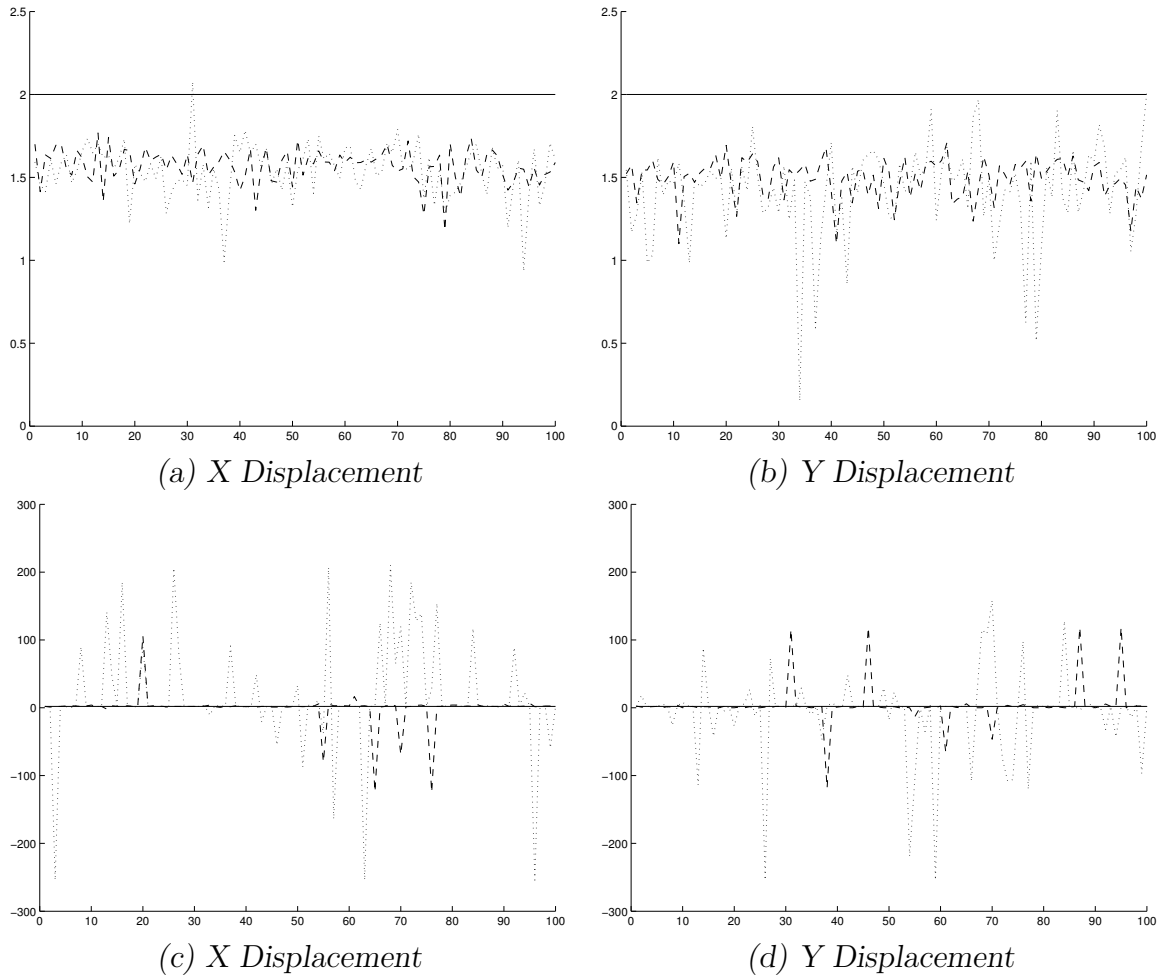


Figure 4.4. Evaluation of pixel shift of two with PSNR of 22 dB. (a) Feature Guided Gautama and Van Hulle algorithm (dashed line) versus Original Gautama and Van Hulle algorithm (dotted line). (b) Feature Guided Gautama and Van Hulle algorithm (dashed line) versus Original Gautama and Van Hulle algorithm (dotted line). (c) Feature Guided Brown algorithm (dashed line) versus Original Brown algorithm (dotted line). (d) Feature Guided Brown algorithm (dashed line) versus Original Brown algorithm (dotted line).

Table 4.1. *Graph Statistics. True displacement is equal to two.*

Algorithm	Mean	Standard Deviation
PSNR = 28dB	X Direction	X Direction
Gautama and Van Hulle	1.7171	0.1067
Feature Guided Gautama and Van Hulle	2.0408	0.1057
Brown	-0.4900	25.6099
Feature Guided Brown	2.0050	0.0500
	Y Direction	Y Direction
Gautama and Van Hulle	1.5460	0.1823
Feature Guided Gautama and Van Hulle	1.8221	0.0998
Brown	-0.7100	25.6051
Feature Guided Brown	1.9400	0.2490
PSNR = 22dB	X Direction	X Direction
Gautama and Van Hulle	1.5404	0.1623
Feature Guided Gautama and Van Hulle	1.5692	0.1022
Brown	-23.3890	76.7416
Feature Guided Brown	-0.5600	23.3890
	Y Direction	Y Direction
Gautama and Van Hulle	1.4135	0.2921
Feature Guided Gautama and Van Hulle	1.4946	0.1186
Brown	-7.7300	60.7299
Feature Guided Brown	3.3250	27.5794

the original algorithms. Therefore, in this section we perform and report on more extensive validations of the Feature Guided Algorithms. The previous test results show that the Feature Guided Guatama and Van Hulle algorithm is more robust to noise deformation. We now evaluate accuracy with respect to pixel shift.

A pixel shift of more than ten is unlikely; therefore, we test shifts from zero to ten. We begin by evaluating the Feature Guided Algorithms over 100 trials, to ensure consistency. No additive White Gaussian Noise is used in these trials. For Figure 4.5 (a) we keep our y displacement at zero while we vary our x displacement from zero to ten in integer values. A total of 1100 trials are executed for this figure. The same technique is used for Figure 4.5 (b), except the y displacement is varied while the x displacement is kept at zero.

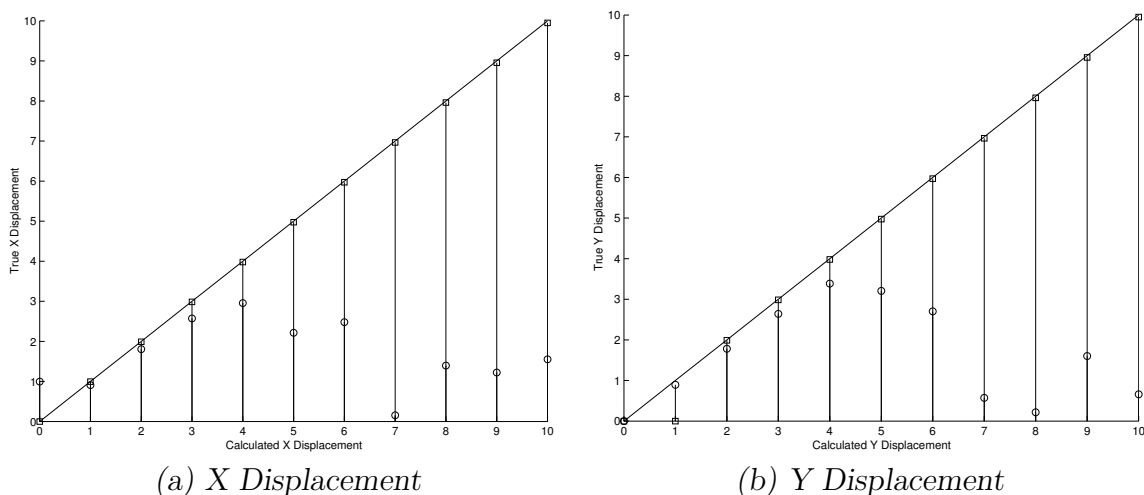


Figure 4.5. *Calculated displacement versus actual displacement. The Feature Guided Brown algorithm is represented by squares. The Feature Guided Gautama and Van Hulle algorithm is represented by circles. The truth data is represented by the line.*

Figure 4.5 shows that the Feature Guided Phase-Based Optic Flow algorithm does not yield efficient estimates of pixel shifts for more than a three pixel displacement. The Feature Guided Brown algorithm is more accurate with respect to large pixel shifts. Table 4.2 shows the results of the displacement tests.

Table 4.2. *Displacement Graphs Statistics.*

True Data(x,y)	Vx	Vy	Tx	Ty
0,0	0.995	0	0	0
1,0	0.907	-0.0059	0.995	0
2,0	1.8052	-0.0161	1.99	0
3,0	2.5713	-0.1989	2.9851	0
4,0	2.9548	-0.2883	3.9801	0
5,0	2.2147	0.2667	4.9751	0
6,0	2.4789	-0.2249	5.9701	0
7,0	-0.1545	0.3165	6.9652	0
8,0	-1.3943	1.3522	7.9602	0
9,0	-1.2227	0.9449	8.9552	0
10,0	-1.5524	-0.9094	9.9502	0
0,1	0.0025	-0.8942	0	0
0,2	0.005	-1.7816	0	1.99
0,3	0.0704	-2.6411	0	2.9851
0,4	0.1207	-3.3851	0	3.9801
0,5	0.2861	-3.2062	0	4.9751
0,6	0.3777	-2.701	0	5.9701
0,7	0.2027	-0.5708	0	6.9652
0,8	-0.0657	-0.2165	0	7.9602
0,9	-0.112	1.601	0	8.9552
0,10	0.0019	0.6594	0	9.9502

4.3.2 Video Stream Data Analysis. As shown in the previous section, the Feature Guided algorithm shows promising results; thus, we evaluate it with video stream data. We test the Feature Guided algorithms against each other using the video stream data. Sixty seconds of the video stream data are evaluated. The large standard deviations of the Feature Guided Brown algorithm, as shown in Table 4.1, now affect the performance. The true estimates and the estimates from the Feature Guided Gautama and Van Hulle algorithm are barely seen due to the inaccurate evaluations from the Brown algorithm as shown in Figure 4.6. Figures 4.7 and 4.8 take a closer look at frames 0 to 15 and 265 to 280 so that we see a set of valid and invalid estimates.

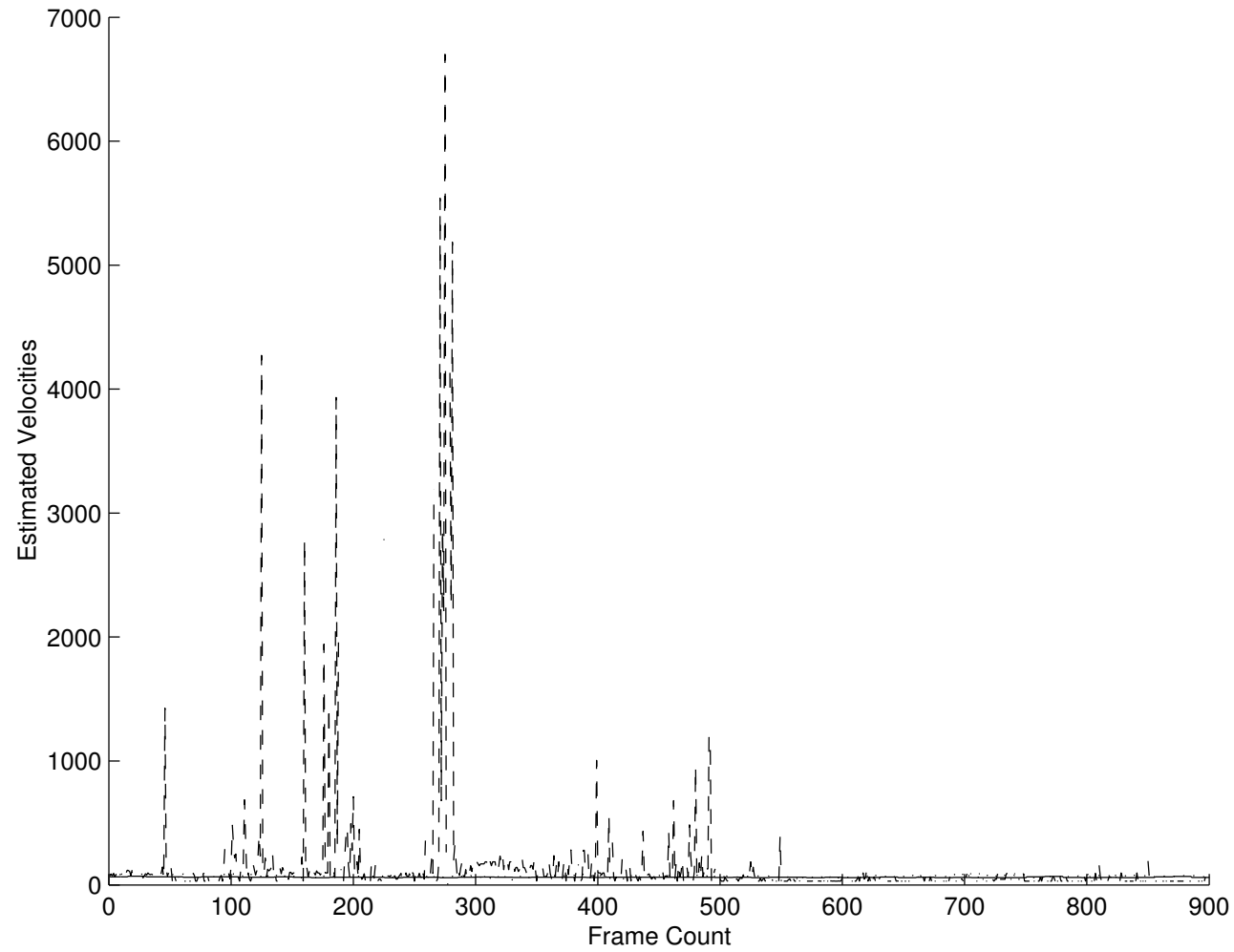


Figure 4.6. *Estimated Velocities.* The truth is represented by the solid line, the Feature Guided Wavelet-Based algorithm is represented by the dashed line, and the Phase-Based Optic Flow algorithm is represented by the dotted line.

4.3.2.1 *Valid Estimates.* For the first second of our video stream data we have a ‘good’ image as defined in Section 3.2.9. We take a closer look at this section so that we can evaluate our algorithm with a ‘good’ image. Figure 4.7 shows the algorithm’s performance over the first 15 frames, which is equal to one second of video data. Our Feature Guided algorithm reveals that we have between seven and twelve valid features in this sequence. The partial list of the features and their calculated velocities are shown in Table 4.4. It is interesting to note that these 15 frames have at least seven valid features, which leads to the hypothesis that one image could be processed to find the regions of interest. These regions could be applied to more than one sequential image, and reasonable results could still be obtained, which would allow our algorithm to be even more computationally effective.

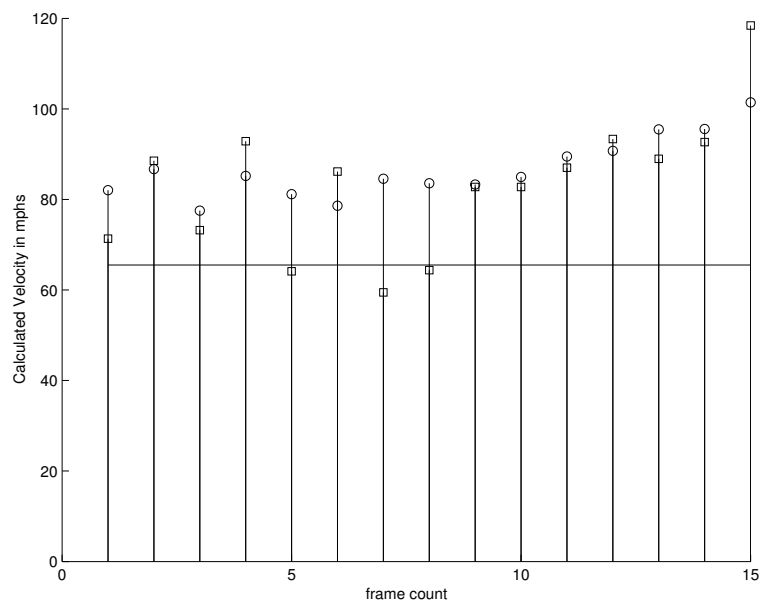


Figure 4.7. *Estimated Velocities.* The truth is represented by the straight line, the Feature Guided Wavelet-Based algorithm is represented by the dashed line, and the Phase-Based Optic Flow algorithm is represented by the dotted line.

Table 4.3. *Valid Estimate Statistics.*

Algorithm	Mean	Stand Deviation
Feature Guided Brown	83.0612	14.9653
Feature Guided Gautama and Van Hulle	86.6872	6.7061

Table 4.4. *Statistics of first second of Video Stream data. Note: Not all features are shown in this table.*

Frame	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7	Mean	σ
0-2	64.8557	64.8557	97.2836	64.8557	64.8557	97.286	97.286	81.07055	17.3344
2-4	64.8557	97.2836	32.4279	97.2836	97.2836	97.2836	133.353	88.53871	31.6847
4-6	61.9023	32.4279	97.2836	64.8557	97.2836	32.4279	97.2836	73.19414	29.51174
6-8	61.9023	97.2836	97.2836	129.7114	97.2836	97.2836	97.2836	92.860925	21.3758
8-10	61.9023	97.2836	97.2836	32.4279	97.2836	97.2836	64.8557	78.33147	25.80499
10-12	61.9023	64.8557	97.2836	97.2836	64.8557	97.2836	97.2836	88.80744	23.29623
12-14	64.8557	64.8557	97.2836	97.2836	32.4279	97.2836	64.8557	72.06192	27.0321
14-16	64.8557	61.9023	97.2836	97.2836	64.8557	32.4279	97.2836	77.53153	22.87979
16-18	61.9023	61.9023	97.2836	129.7114	32.4279	102.0885	64.8557	82.74877	29.44232
18-20	61.9023	61.9023	97.2836	97.2836	64.8557	64.8557	97.2836	82.74877	18.46803
20-22	64.8557	64.8557	97.2836	98.2836	64.8557	102.0885	97.2836	87.11929	16.76669
22-24	61.9023	97.2836	97.2836	129.7114	64.8557	97.2836	97.2836	93.35233	20.04954
24-26	61.9023	129.7114	129.7114	64.8557	97.2836	129.7114	102.0885	98.02317	28.6158
26-28	64.8557	97.2836	97.2836	64.8557	129.7114	129.7114	97.2836	97.2836	24.51315
28-30	71.8626	162.1393	129.7114	97.2836	129.7114	129.7114	129.7114	118.4268	27.98583

Further evaluation of our Feature Guided Brown algorithm is needed. All constraints and error correction in the algorithm are based on Gautama and Van Hulle [10], which is an appropriate approach since we want to ensure a valid comparison. However, we know that the Feature Guided Brown algorithm is capable of making extremely wrong estimates, due to its correlation process as defined in Section 3.3.5. We can compensate for this possibility by using our standard deviation statistic.

4.3.2.2 Invalid Estimates with velocity correcting Code. The Feature Guided Brown algorithm proves to be either very accurate or extremely inaccurate. A simple velocity correcting code could be added to increase the performance of the Feature Guided Brown algorithm. The correcting code could use the standard deviation of the velocities as its error metric. If we decrease the standard deviation, we may be able to increase the accuracy; however, this decreases our ability to interpret different topographical data. Therefore, we chose to apply our correcting code to the averaged velocities instead of the feature velocities. For these reasons, more analysis should be performed. We seek to determine if this approach improves the performance of the Feature Guided Brown algorithm. From Figure 4.6 it is obvious that frames 265 to 280 have extremely poor estimates. Figure 4.8 provides a closer look.

Table 4.5. *Invalid Estimate Statistics.*

Algorithm	Mean	Stand Deviation
Feature Guided Brown	1763.2	2190.2
Feature Guided Gautama and Van Hulle	55.0139	20.3177

Our Velocity Correcting code assumes that the initial five estimates are correct. Next, we determine the standard deviation for each frame. If it is above 10 we take the mean of the previous five estimates and use that as our value. The improvement for frames 268 to 280 is shown in Figure 4.9 and Table 4.6. The overall improvements

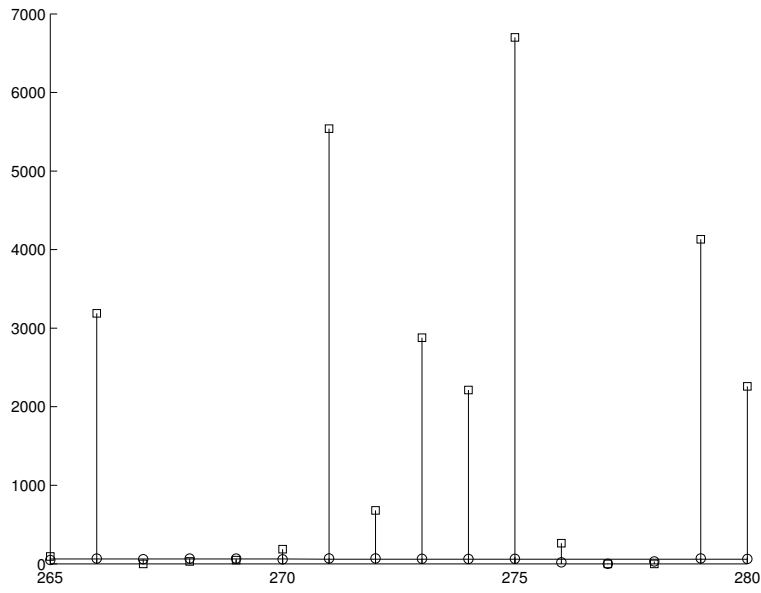


Figure 4.8. *Estimated Velocities for Frames 265 to 280. The Feature Guided Brown algorithm is represented by squares, the Feature Guided Gautama, and Van Hulle algorithm is represented by circles, and the GPS data is represented by the straight line.*

are shown in Figure 4.10. The mean and standard deviation calculations are greatly improved with the use of this velocity correction code.

Table 4.6. *Invalid Data Corrected Statistics.*

Algorithm	Mean	Stand Deviation
Feature Guided Brown	92.9687	2.9354e-014
Feature Guided Gautama and Van Hulle	55.0139	20.3177

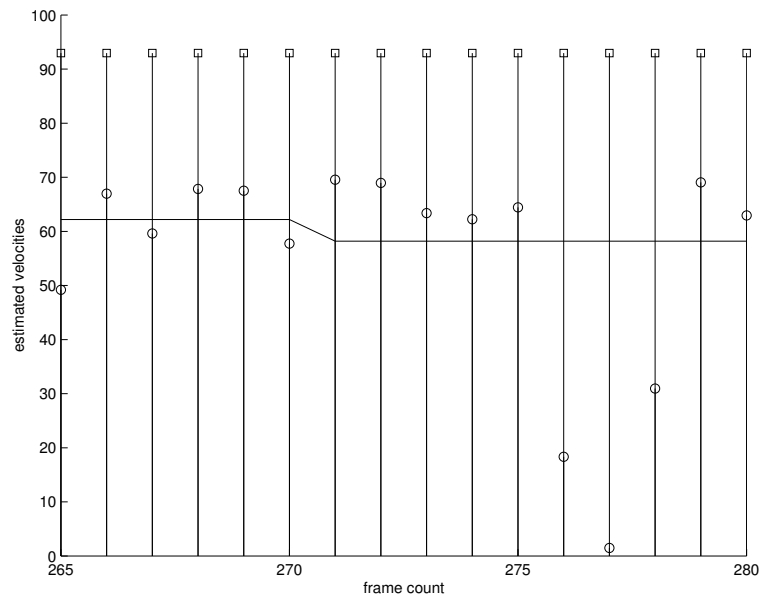


Figure 4.9. *Estimated Velocities for Frames 265 to 280. The Feature Guided Brown algorithm is represented by squares, the Feature Guided Gautama and Van Hulle algorithm is represented by circles, and the GPS data is represented by the straight line.*

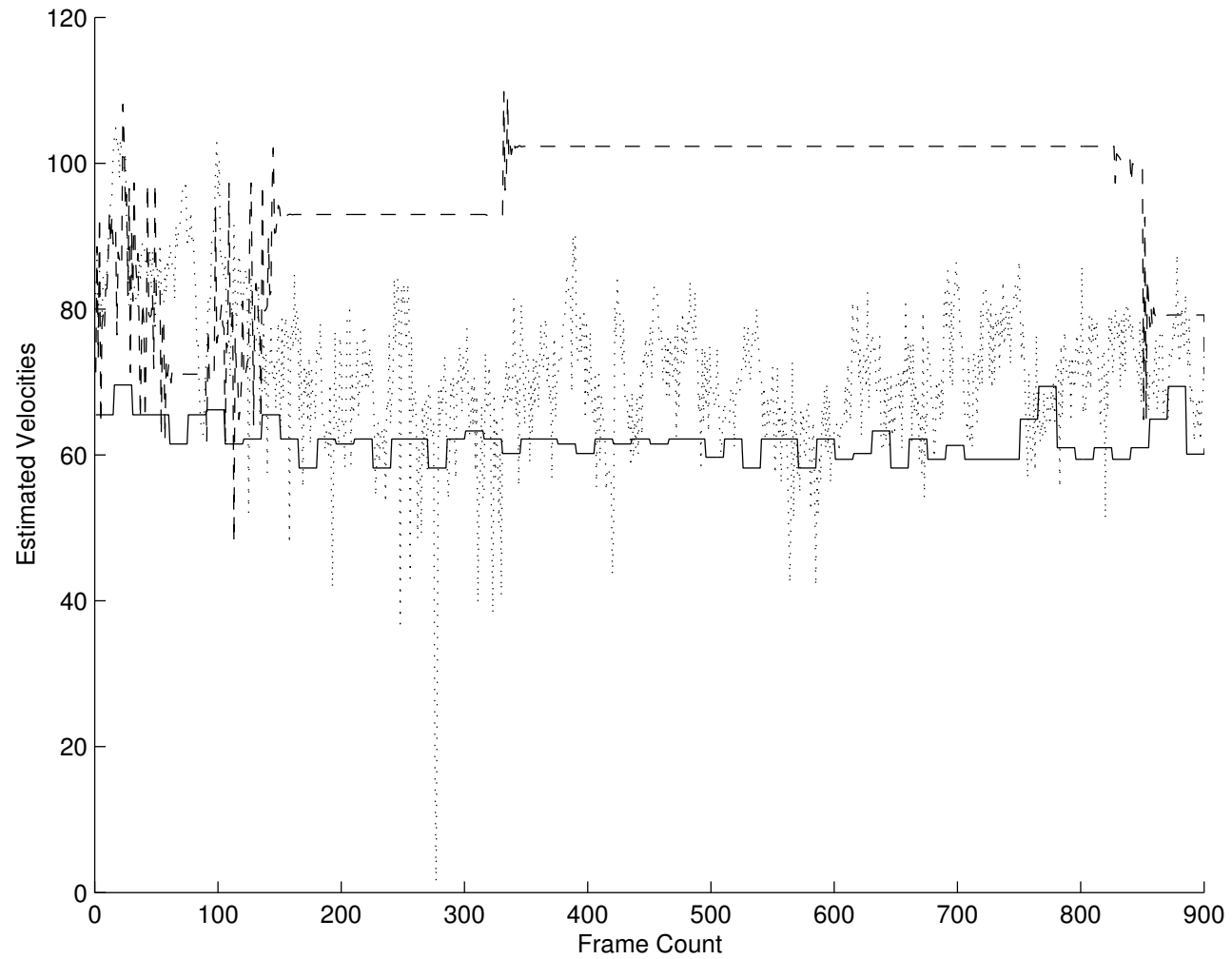


Figure 4.10. *Estimated Velocities. The Feature Guided Brown algorithm is represented by the dashed line, the Feature Guided Gautama, and Van Hulle algorithm is represented by the dotted line. The GPS data is represented by the solid line.*

4.3.2.3 Automation. As stated in Chapter Three, our algorithm determines when there is not enough texture present to guarantee valid estimates of pixel shifts. Frames where an inaccurate estimate may have taken place are analyzed. The Feature Guided Brown algorithm yields a standard deviation of 281.5818, and the Gautama and Van Hulle algorithm yields a standard deviation of 8.7640. We expect to have a standard deviation less than 3 due to our truth data. These calculated standard deviations are far worse than our statistics presented in Table 4.4. Approximately 30 of the 379 uncertain calculated velocities for the Feature Guided Brown algorithm fall within 10 miles per hour of the calculated mean velocity, 65.5280. Approximately 258 of the 379 uncertain calculated velocities for the Feature Guided Gautama and Van Hulle algorithm fall within 10 miles per hour of the calculated mean velocity, 70.8236. This indicates that the images that are determined not to have enough texture do not yield accurate estimates.

4.4 Summary

Overall, we provide a validation study that shows the increased accuracy and robustness of our Feature Guided algorithms. For the synthetic data tests, the Feature Guided Brown algorithm estimates more accurate pixel displacement. However, it lacks the robustness to noise that the Feature Guided Gautama and Van Hulle algorithm displays. The Feature Guided Brown Algorithms also lacks the accuracy of the Feature Guided Gautama and Van Hulle algorithm with respect to the Video Stream Data. This result is expected since the Feature Guided Brown algorithm can only estimate integer values.

The automation criteria prove to be effective for providing a valid confidence measurement for accuracy. However, the Feature Guided Brown algorithm has additional accuracy problems, with respect to the video stream data, even with the velocity correcting code of Section 4.3.2.2. Therefore, it is important to understand the requirements of the application and choose the appropriate optic-flow algorithm.

V. Discussion and Future Work

5.1 Contributions of this Thesis

In this thesis, we improve efficiency, robustness, and computational efficiency of preexisting Phase- and Wavelet-Based Algorithms by adding our Feature Guided algorithm. Through an innovative technique, we define image ‘texture’, so that we can determine the validity of our estimates. Through the use of the redundant discrete wavelet transform, binary thresholding and masking, and morphological imaging, we decompose a wide range of images into subimages based upon the feature locality. The Feature Guided Brown algorithm proves to be more accurate with respect to integer pixel displacement. However, it lacks accuracy for non-integer pixel shifts and robustness to noise that the Feature Guided Gautama and Van Hulle algorithm displays. Recall from Section 1.1 that the long term goal of this research effort is to improve the accuracy and efficiency of optic flow algorithms, so that they may be used for Air Force navigational systems. Based on the results presented, the Feature Guided algorithm has moved us closer to reaching our goal.

5.2 Potential for Future Research

Even though our algorithm realizes significant gains in the optic flow algorithm’s accuracy and efficiency, there are still many aspects that must be resolved before we can use this technique in navigation. The topics presented below are areas that need to be researched.

5.2.1 Other Image Registration Algorithms. Our Feature Guided algorithm could provide improved accuracy and efficiency for many other image registration algorithms. For example, it could be implemented as a front-end to a super-resolution algorithm. Most super-resolution algorithms are interested in specific objects in an

image. Our feature guided algorithm can find and register these objects of interest, thus significantly reducing the computational load on the super-resolution algorithm.

Our algorithm could also be directly applied to other image registration algorithms, such as Manfra’s algorithm [23]. Manfra’s algorithm uses the continuous wavelet transform to compute more accurate translations, however, efficiency is sacrificed. The combination of the Feature Guided algorithm and Manfra’s algorithm [23] could increase registration accuracy while significantly decreasing computational complexity.

5.2.2 Use of Subbands. Currently our algorithms use the wavelet subband to find regions of interest (ROI). Those ROIs are then imposed on the original images. One variation of our algorithm would be to impose those ROIs on the subbands themselves. This would increase the computational efficiency of our translation estimates. Since the High-Low Subband preserves vertical edges, it would be ideal for estimating displacements in the x direction. The Low-High Subband could then be used to determine the displacements in the y direction. Since the subbands contains less information than the original image, it would be less computationally expensive to compare the respective subbands to each other.

5.2.3 Finding Redundant Objects. The Feature Guided algorithm determines where a feature is located. It is logical to assume that if it is located in the middle of an image, then it will also occur in the next series of frames. With this information, it is possible to define “redundant features” as those likely to appear in the next set of frames. This a priori information on feature location can be used instead of redefining the feature locations for each image. This procedure would be more computationally effective than our current Feature Guided Algorithm, which only imposes the regions of interest on one image and begins a new feature for every frame.

5.2.4 Dealing with Video Stream Data Rotations. To reach our long term goal of having an image registration algorithm sufficiently efficient and accurate to use in a navigation system, we must take platform rotations into account. Traditional registration algorithms do not account for the aircraft rotations that are likely to occur. The objects on the ground may have distorted depths and velocities due to aircraft maneuvers. We currently use the estimates from multiple features to form a single velocity estimate. However, if the sensor platform is rotating, different features should be at different velocities. The individual features estimates could be combined to estimate the rotations of the aircraft.

Bibliography

1. Allen, R. L., et al. "Laplacian and Orthogonal Wavelet Pyramid Decompositions in Coarse-to-Fine Registration," *IEEE Transactions on Signal Processing*, 41(12):3536–3541 (December 1993).
2. Aloimonos, Y. and Z. Duric. "Active egomotion estimation: a qualitative approach." *European Conference on Computer Vision* 2. 497–510. 1992.
3. Barron, J. L., et al. "Performance of Optic Flow Techniques," *International Journal of Computer Vision*, 12:43–77 (1994).
4. Brown, L. G. "A Survey of image Registration Techniques," *ACM Computing Surveys*, 24(4):325–376 (December 1992).
5. Brown, R. K. *Image Registration using Redundant Wavelet Transforms*. MS thesis, Air Force Institute of Technology, Wright Patterson AFB, OH, March 2001.
6. Corvi, M. and G. Nicchiotti. "Multiresolution Image Registration." *Proceedings of IEEE International Conference on Image Processing*. 224–227. October 23-26 1995.
7. Fleet, David J. "Stability of Phase Information." *IEEE transaction on Pattern Analysis and Machine Intelligence* 15. 1253–1268. December 1993.
8. Fleet, D.J. *Measurement of Image Velocity*. Kluwer Academic Publishers, Norwell, 1992.
9. Fleet, D.J. and A.D. Jepson. "Computation of component image velocity from local phase information," *International Journal of Computer Vision*, 5:77–104 (1990).
10. Gautama, Temujin and Marc M. Van Hulle. "A Phased-Based Approach to the Estimation of the Optical Flow Field Using Spatial Filtering." *IEEE Transactions on Neural Networks* 13. 1127–1136. September 2002.
11. Guo H., et al. "Wavelet Based Speckle Reduction with Applications to SAR based ATD/R." *IEEE International Conference on Image Processing*. Nov 1994.
12. Hangche Liu, et. al. "Accuracy vs. Efficiency Trade-offs in Optic Flow Algorithms," *Computer Vision and Image Understanding*, 72:217–286 (December 1998).
13. Heeger, D.J. "Optic Flow using spatiotemporal filters," *International Journal of Computer Vision*, 1:279–302 (1981).
14. Horn, B.K.P. and B.G. Schunck. "Determining Optical Flow," *Artificial Intelligence*, 16:185–203 (August 1981).

15. LeMoigne, J. "Parallel Registration of Multi-Sensor Remotely Sensed Imagery Using Wavelet Coefficients." *OE/Aerospace Sensing, Wavelet Applications Conference, 2242*. 432–443. April 5-8 1994.
16. LeMoigne, J., et al., "An Automated Parallel Image Registration Technique of Multiple Source Remote Sensing Data." *CESDIS TR-96-182* - submitted to IEEE Transactions in Geoscience and Remote Sensing.
17. LeMoigne, J. and R. F. Crompt. "The Use of Wavelets for Remote Sensing Image Registration and Fusion." *Wavelet Applications, 2762*. 535–544. 1996.
18. LeMoigne, J. and I. Zavorin. "An Application of Rotation- and Translation-Invariant Overcomplete Wavelets to the Registration of Remotely Sensed Imagery." *SPIE's OE/Aerospace Sensing, Wavelet Applications Conference*. 130–140. April 6-8 1999.
19. LeMoigne, J. and I. Zavorin. "Use of Wavelets for Image Registration." *Wavelet Applications VII 4056*, edited by H. Szu, et al. 99–108. 2000.
20. Li, H. H. and Y. Zhou. "A Wavelet-Based Feature Extractor for Multi-Sensor Image Registration." *Proceedings of the SPIE, 2762*. 524–534. 1996.
21. Lucas, B.D. *Generalization Image Matching by Differences*. PhD dissertation, Carnegie-Mellon University, Pittsburg, PA, 1984.
22. Lucas, B.D. and T. Kanade. "An interactive image-registration technique with application to stereo vision." *DARPA Image Understanding Workshop*. 121–130. 1981.
23. Manfra, Jennifer L. *Translation and Rotation Invariant Multiscale Image Registration*. MS thesis, Air Force Institute of Technology, Wright Patterson AFB, OH, March 2002.
24. Manjunath, B. S., et al. "A New Approach to Image Feature Detection with Applications," *Elsevier Science Pattern Recognition, 29(4)*:627–640 (1996).
25. Nagel, H.H. "On the estimation of optic flow: Realations between different approaches and some new results," *Artificial Intelligence, 33*:299–324 (1987).
26. Sharman, R., et al. "A Fast and Accurate Way to Register Medical Images Using Wavelet Modulus Maxima," *Elsevier Science Pattern Recognition Letters, 21*:447–462 (2000).
27. Simoncelli, E.P. *Distributed Representation and Analysis of Visual Motion*. PhD dissertation, Massachusetts Institute of Technology, Cambridge, MA, January 1993.
28. Simoncelli, E.P., et al. "Probability distributions of optic flow." *IEEE Proceedings of Computer Vision and Pattern Recognition*. 310–315. June 1991.

29. Tashakkori, R., et al. "Prediction of Medical Images Using Wavelets." *Wavelet Applications VII*, 4056, edited by H. Szu, et al. 332–340. 2000.

Vita

Kate R. Duffy graduated from the Northeastern University in 2001 with a B.S. in Electrical Engineering. She was selected as a direct assessment to the Air Force Institute of Technology (AFIT) at Wright Patterson AFB, OH where she received her M.S. in Electrical Engineering in March of 2003. Lieutenant Duffy's next assignment will be to the Air Force Research Laboratory at Wright Patterson Air Force Base, where she will be working in the Sensors Directorate with the Implementation and Assessment Branch.

REPORT DOCUMENTATION PAGE

*Form Approved
OMB No. 074-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 25-03-2003		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From - To) Jun 2002 - Mar 2003	
4. TITLE AND SUBTITLE FEATURE GUIDED IMAGE REGISTRATION APPLIED TO PHASE- AND WAVELET-BASED OPTIC FLOW				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Duffy, Kate, R., 2d Lieutenant, USAF				5d. PROJECT NUMBER If funded, enter ENR #	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/ENG) 2950 Hobson Way, Building 640 WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GE/ENG/03-09	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/MNGN Attn: Maj. Ryan Pendleton 101 W. Eglin Blvd Ste 152 Eglin AFB FL 32542-6810				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Optic Flow algorithms are useful in problems such as computer vision, navigational systems, and robotics. However, current algorithms are computationally expensive or lack the accuracy to be effective compared with traditional navigation systems. Recently, lower accuracy inertial navigation systems (INS) based on Microelectromechanical systems (MEMS) technology have been proposed to replace more accurate traditional navigation systems. An Optic Flow algorithm can be created that is, unlike GPS, not susceptible to jamming or spoofing. Our long term goal is to use a robust Optic Flow algorithm in conjunction with a MEMS INS to create a more accurate and less expensive MEMS INS navigational system. We propose a new wavelet-based technique for motion analysis based on feature extraction. With the use of the redundant discrete wavelet transform and morphological processing, dominant image features are efficiently extracted from the image, allowing for a robust and computationally effective optic flow algorithm. These features are used to guide the registration; the image is broken into subsections based on the spatial location and extent of the detected features. These subsections are incorporated into optic flow algorithms to create local estimates of image motion. We verify the accuracy of our algorithm by analyzing actual aircraft video with known position and inertial references.					
15. SUBJECT TERMS Wavelets Transforms, Image Registration, Motion					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Roger L. Claypoole, Maj, USAF (ENG)
U	U	U	UU	75	19b. TELEPHONE NUMBER (Include area code) (937) 255-3636, ext 4625; e-mail: Roger.Claypoole@afit.edu