

SUBPIXEL ACCURACY IMAGE REGISTRATION  
WITH APPLICATION TO UNSTEADINESS  
CORRECTION

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND  
ELECTRONICS ENGINEERING  
AND THE INSTITUTE OF ENGINEERING AND SCIENCES  
OF SIKKENT UNIVERSITY  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
MASTER OF SCIENCE

By

Çiğdem Eroğlu

January 1997

TA  
1632  
E76  
1997

SUBPIXEL ACCURACY IMAGE REGISTRATION  
WITH APPLICATION TO UNSTEADINESS  
CORRECTION

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND  
ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES  
OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
MASTER OF SCIENCE

By

*Çiğdem Eroğlu.*

Çiğdem Eroğlu

January 1997

TH  
1632  
.E76  
1997

B 196713

I certify that I have read this thesis and that in my opinion it is fully adequate,  
in scope and in quality, as a thesis for the degree of Master of Science.

  
Assist. Prof. Dr. Tanju Erdem (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate,  
in scope and in quality, as a thesis for the degree of Master of Science.

  
Prof. Dr. Levent Onural

I certify that I have read this thesis and that in my opinion it is fully adequate,  
in scope and in quality, as a thesis for the degree of Master of Science.

  
Assist. Prof. Dr. Orhan Arıkan

Approved for the Institute of Engineering and Sciences:

  
Prof. Dr. Mehmet Baray  
Director of Institute of Engineering and Sciences

## ABSTRACT

### SUBPIXEL ACCURACY IMAGE REGISTRATION WITH APPLICATION TO UNSTEADINESS CORRECTION

Çiğdem Eroğlu

M.S. in Electrical and Electronics Engineering

Supervisor: Assist. Prof. Dr. Tanju Erdem

January 1997

Image registration refers to the problem of spatially aligning the images in an image sequence. The proposed efficient search method estimates the subpixel displacements causing the misregistration of two frames faster than other methods without any loss in accuracy. It is assumed that the misregistration is due to global motion. The criterion function used is the mean-squared error over the displaced frames in which image intensities at subpixel locations are evaluated using bilinear interpolation only once in the formula. A novel near-closed-form solution does not use any search unless absolutely necessary. An extension of the near-closed-form solution that is insensitive to intensity variations between the frames can account for contrast and brightness changes in a sequence. Simulations on unsteady image sequences demonstrate the superiority of the proposed near closed-form solution. The application to de-interlacing also gives good results.

*Keywords:* Image registration, subpixel accuracy, subpixel registration, motion estimation, displacement estimation, sequence stabilization, unsteadiness correction.

## ÖZET

### KESİRLİ İMGE ÖĞESİ DOĞRULUĞUNDA İMGE HİZALAMA VE BUNUN TİTREK İMGE DİZİLERİNİN SABİTLENMESİNDE UYGULANMASI

Çiğdem Eroğlu

Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans

Tez Yöneticisi: Yrd. Doç. Dr. Tanju Erdem  
Ocak 1997

İmge hizalama, bir imge dizisindeki imgelerin karşılıklı noktalar denk gelecek şekilde çakıştırılması problemi. Önerilen yeni bir arama yöntemi, iki imge arasındaki kesirli imge ögesi boyutundaki kaymaları, diğer yöntemlere göre daha hızlı ve hassasiyet kaybı olmadan bulmaktadır. Kaymanın, imge üzerindeki her noktada aynı olduğu varsayılmıştır. Kriter olarak kullanılan fonksiyon, dayanak imge ile düzeltilmiş imge arasındaki hata karelerinin ortalaması olup, kesirli imge noktacılarındaki değerler ise, çiftdoğrusal aradeğerleme ile, formülde bir kez kullanarak bulunmuştur. Önerilen kapalı bir çözüm ise, zorunlu olmadıkça arama kullanılmamaktadır. Bu kapalı çözüm, iki imge arasındaki ışık değişimlerine duyarsız olacak biçimde geliştirilmiştir. Bu kapalı çözümün titretilen imge dizilerine uygulanması, bu yöntemin diğer metotlara göre daha iyi olduğunu göstermiştir.

*Anahtar Kelimeler:* İmge hizalama, imge çakıştırma, kesirli imge ögesi doğruluğunda hizalama, hareket kestirimi, kayma kestirimi, titretilen imge dizilerinin sabitlenmesi.

## ACKNOWLEDGEMENT

I gratefully thank my supervisor Assist. Prof. Dr. Tanju Erdem for his suggestions, supervision and guidance throughout the development of this thesis.

I would also like to thank Prof. Dr. Enis Çetin for his suggestions and guidance; Assist. Prof. Dr. Orhan Arıkan and Prof. Dr. Levent Onural, the members of my jury, for reading and commenting on the thesis.

Many thanks to Tunç Bostancı for his help in the recording processes and all my friends for their valuable discussions, help and friendship.

It is a pleasure to express my special thanks to my mother, father and brother for their sincere love, support and encouragement.

# TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	Literature Review	4
1.2.1	Search Techniques	5
1.2.2	Correlation Techniques . . . . .	6
1.2.3	Differentiation Techniques . . . . .	8
1.2.4	Feature Matching Techniques and Others . . . . .	9
1.3	Contribution and Scope	10
<b>2</b>	<b>AN EFFICIENT SEARCH METHOD</b>	<b>12</b>
2.1	Background . . . . .	12
2.2	Method	14
2.3	Results . . . . .	16

”

<b>3</b>	<b>A NEAR-CLOSED-FORM SOLUTION</b>	<b>20</b>
3.1	Method	20
3.2	Results . . . . .	22
3.2.1	Simulations with the synthetic Text Sequence . . . . .	23
3.2.2	Simulations with the CT Sequence	24
3.2.3	Simulations with the Bilkent Sequence . . . . .	25
<b>4</b>	<b>ACCOUNTING FOR INTENSITY VARIATIONS</b>	<b>37</b>
4.1	Modeling . . . . .	37
4.2	Method	38
4.3	Results . . . . .	41
4.3.1	Text-1 Sequence . . . . .	42
4.3.2	Text-2 Sequence . . . . .	43
4.3.3	Text-3 Sequence . . . . .	44
4.3.4	Text-4 Sequence . . . . .	44
4.3.5	CT-2 Sequence . . . . .	45
4.3.6	Bilkent-2 Sequence . . . . .	45
4.3.7	Summary of Results	46
<b>5</b>	<b>CONCLUSIONS AND FUTURE WORK</b>	<b>61</b>
	<b>APPENDICES</b>	<b>64</b>
<b>A</b>	<b>PHASE CORRELATION ALGORITHM</b>	<b>65</b>

<b>B EXISTING SUBPIXEL DISPLACEMENT ESTIMATION METHODS</b>	<b>67</b>
B.1 Phase-correlation and Cross-correlation Surface Interpolation . .	67
B.2 Logarithmic Search Method . . . . .	69
B.3 Differentiation Method . . . . .	70
<b>C FORMULAS USED IN THE NEAR-CLOSED-FORM SOLUTION</b>	<b>72</b>
C.1 Basic Summations	72
C.2 MSE Coefficients . . . . .	73
C.3 The Coefficients of the Fifth Order Polynomial . . . . .	74
C.4 The Coefficients for Contrast and Brightness Parameters . . . .	74
C.5 MSE Coefficients in the Case of Intensity Variations . . . . .	75
C.6 The Coefficients for the Brightness Parameter . . . . .	77
C.7 The Coefficients for the Contrast Parameter . . . . .	77

\*

## LIST OF FIGURES

1.1	Effects of the horizontal translational movement of the camera or the film during digitization of the film. . . . .	3
2.1	The bilinear interpolation for positive $d_1, d_2$ . . . . .	13
2.2	Exhaustive search . . . . .	14
2.3	The first frame of the Text sequences. . . . .	17
3.1	The first frame of the Text Sequence for 5 dB SNR. . . . .	23
3.2	Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for Text Sequence with 20 dB SNR. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for $d_1$ , while the second column compares the absolute errors for $d_2$ .	28

3.3	Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for Text Sequence with 10 dB SNR. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for $d_1$ , while the second column compares the absolute errors for $d_2$ .	29
3.4	Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for Text Sequence with 5 dB SNR. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for $d_1$ , while the second column compares the absolute errors for $d_2$ . . . . .	30
3.5	The first frame of the CT Sequence. . . . .	31
3.6	Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for CT Sequence. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for $d_1$ , while the second column compares the absolute errors for $d_2$ . . . . .	32
3.7	Six frames of the Bilkent Sequence: first and fifth frames (top row), ninth and thirteenth frames (second row), seventeenth and twentieth frames (third row).	33

#

3.8	Pixel part of the displacements obtained by the phase correlation method for the “Bilkent” Sequence. . . . .	34
3.9	Plots of subpixel part of the displacements of the near-closed-form solution together with the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for “Bilkent” Sequence. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares $d_1$ , while the second column compares $d_2$ . . . . .	35
3.10	The figure at the top shows a frame of the Bilkent Sequence which is degraded due to the motion of the camera. The figure at the bottom shows the same frame after motion compensation of the odd field with respect the even field using the near closed-form solution. . . . .	36
4.1	(a) The reference frame of all Text Sequences. (b) The last frame of the Text-2 Sequence. (c) The last frame of the Text-3 Sequence. (d) The last frame of the Text-4 Sequence	47
4.2	The last frame of the CT-2 Sequence generated with $(\gamma_0, \eta_0) = (-0.02, -2)$ . . . . .	47
4.3	Six frames of the Bilkent-2 Sequence which contains intensity variations. First and seventh frames (top row), thirteenth and fifteenth frames (second row), seventeenth and twentieth frames (third row). . . . .	48

4.4	$(\gamma_0, \eta_0) = (0, 0)$ . Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for Text-1 Sequence with 10dB SNR. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for $d_1$ , while the second column compares the absolute errors for $d_2$ . . . . .	49
4.5	The $\gamma$ and $\eta$ values found for the Text-1 Sequence which does not contain any intensity variations. The sign “star” and “+” signs denote the estimated and true values for $\gamma$ (on the left) and $\eta$ (on the right) respectively.	50
4.6	$(\gamma_0, \eta_0) = (-0.02, -2)$ . Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for Text-2 Sequence with 10dB SNR. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for $d_1$ , while the second column compares the absolute errors for $d_2$ . . . . .	51
4.7	The $\gamma$ and $\eta$ values found for the Text-2 Sequence in which $(\gamma_0, \eta_0) = (-0.02, -2)$ . The “*” and “+” signs denote the estimated and true values for $\gamma$ (on the left) and $\eta$ (on the right) respectively.	52

v

- 4.8  $(\gamma_0, \eta_0) = (-0.04, 0)$ . Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for Text-3 Sequence with 10dB SNR. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for  $d_1$ , while the second column compares the absolute errors for  $d_2$ . . . . . 53
- 4.9 The  $\gamma$  and  $\eta$  values found for the Text-3 Sequence in which  $(\gamma_0, \eta_0) = (-0.04, 0)$ . The “\*” and “+” signs denote the estimated and true values for  $\gamma$  (on the left) and  $\eta$  (on the right) respectively. 54
- 4.10  $(\gamma_0, \eta_0) = (0, -3)$ . Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for Text-4 Sequence with 10dB SNR. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for  $d_1$ , while the second column compares the absolute errors for  $d_2$ . . . . . 55
- 4.11 The  $\gamma$  and  $\eta$  values found for the Text-3 Sequence in which  $(\gamma_0, \eta_0) = (0, -3)$ . The “\*” and “+” signs denote the estimated and true values for  $\gamma$  (on the left) and  $\eta$  (on the right) respectively. 56

”

4.12	$(\gamma_0, \eta_0) = (-0.02, -2)$ . Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for CT-2 Sequence. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for $d_1$ , while the second column compares the absolute errors for $d_2$ . . . . .	57
4.13	The $\gamma$ and $\eta$ values found for the CT-2 Sequence in which $(\gamma_0, \eta_0) = (-0.02, -2)$ . The “*” and “+” signs denote the estimated and true values for $\gamma$ (on the left) and $\eta$ (on the right) respectively.	58
4.14	Pixel part of the displacements obtained by the phase correlation method for the “Bilkent-2” Sequence. . . . .	58
4.15	Plots of subpixel part of the displacements of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for “Bilkent-2” Sequence. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares $d_1$ , while the second column compares $d_2$ . . . . .	59
4.16	The $\gamma$ and $\eta$ values found for the Bilkent-2 Sequence. . . . .	60
A.1	The phase-correlation function for a $40 \times 40$ block of two images which are relatively shifted.	66
B.1	The phase-correlation surface interpolation points. . . . .	68
B.2	Logarithmic (three-step search) . . . . .	70

”

## LIST OF TABLES

2.1	The random displacements of the frames in the Text sequence. The first frame is the reference frame and has frame number 1. $d_1$ and $d_2$ denote the vertical and horizontal displacement, respectively. . . . .	18
2.2	The mean of absolute displacement errors ( $\epsilon(d_1)$ and $\epsilon(d_2)$ ) and the standard deviation of absolute displacement errors ( $\sigma(d_1)$ and $\sigma(d_2)$ ) found by the proposed search, exhaustive search and logarithmic search methods for 1/8 and 1/16 pixel search accuracies. . . . .	19
2.3	CPU times (in seconds) for the three search-based methods. Case 1: A: 1/8 pixels, B:(50 × 50), Case 2: A: 1/8 pixels, B:(100 × 50), Case 3: A: 1/16 pixels, B:(50 × 50), and Case 4: A: 1/16 pixels, B:(100 × 50) where A denotes the accuracy and B denotes the block size.	19
3.1	The mean absolute displacement estimation errors $\epsilon(d_1)$ and $\epsilon(d_2)$ for Text Sequence. Block size is 130 × 60, CPU time (in seconds) includes the time spent for the detection of the pixel movement using the phase correlation algorithm. . . . .	31

7

3.2	The mean absolute displacement estimation errors $\epsilon(d_1)$ and $\epsilon(d_2)$ over the 19 frames for the CT Sequence. Block size is $60 \times 60$ , CPU time (in seconds) includes the time spent for the detection of the pixel movement using the phase correlation algorithm. . . . .	31
4.1	The mean absolute displacement estimation errors $\epsilon(d_1)$ and $\epsilon(d_2)$ .	46
4.2	The mean absolute displacement estimation errors $\epsilon(d_1)$ and $\epsilon(d_2)$ .	46
4.3	The mean absolute displacement estimation errors $\epsilon(d_1)$ and $\epsilon(d_2)$ for the CT-2 Sequence containing intensity variations. . . .	47

**To my parents . . .**

# Chapter 1

## INTRODUCTION

Image registration refers to the problem of spatially aligning the images in an image sequence with respect to a reference frame [1, 2]. Misregistration of images may result from translational motion or more complex spatial transformations such as rotation and scaling. These spatial transformations may also vary locally. In this thesis, a global translational motion, i.e., a global displacement of the frames, is considered as the cause of misregistration.

The displacement of a frame in general has a pixel part as well as a subpixel part. In this thesis, a novel near-closed-form solution to the estimation of the subpixel part of the displacement of a frame is proposed. The application of the proposed method to the stabilization of unsteady image sequences is presented. An extension of the method that is insensitive to intensity variations, i.e., illumination effects, is also proposed. Simulation results on unsteady image sequences are given to demonstrate the superiority of the proposed near closed-form solution to the existing subpixel displacement estimation methods in the literature.

The rest of Chapter 1 gives the motivation to register images. Then, a comprehensive survey of image registration literature is given. Finally, the contribution and scope of this thesis is presented.

Chapter 2 introduces a new efficient search method for finding the subpixel displacement. In Chapter 3, the proposed near closed-form solution is presented and experimental results are given to evaluate its performance. Chapter 4 extends the method of Chapter 3 to sequences containing intensity variations. Chapter 5 gives the conclusions and future directions for research.

## 1.1 Motivation

The research that resulted in this thesis is motivated by the enormity of applications that employ subpixel image registration. One of the most common applications of subpixel image registration is in the stabilization of unsteady image sequences. Image unsteadiness may be caused by any unwanted and unpredictable relative movements of a camera and a scene during the recording of the scene, or of a scanner and a motion picture film during the digitization of the film for movie post-production. In such applications, image registration problem refers to estimation of the global displacement of each frame in the sequence with respect to a reference frame, and then spatially shifting back each frame with the estimated displacement. It is necessary to estimate these relative displacements down to subpixel accuracy, because subpixel translations in a sequence may cause a disturbing jitter, especially in stationary scenes. Figure 1.1 illustrates the effects of the horizontal subpixel movement of the camera or of the film (which is held by the pins) during the digitization of the film. As can be seen in Figure 1.1, the sample points are horizontally shifted with respect to each other.

The displacement of a frame in general has a fractional (i.e., subpixel) part as well as an integer (i.e., pixel) part. The integer part of the displacement can be found using one of the well-known techniques in [1], such as the phase correlation technique [3]. In this thesis, our aim is to estimate the subpixel part of the displacement between similar images given its pixel part.

Subpixel image registration is also needed in data comparison for detection and monitoring of changes. Data comparison is frequently employed in medical image analysis. For example, in digital subtraction angiography (DSA) [4, 5, 6],

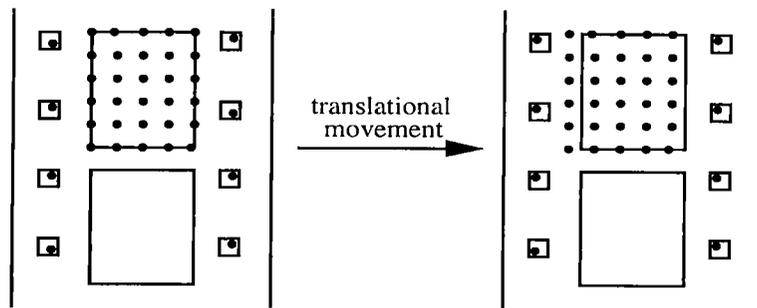


Figure 1.1: Effects of the horizontal translational movement of the camera or the film during digitization of the film.

digital subtraction mammography [1], and X-ray imaging [7, 8], registration of images taken before and after radio isotope injection is required so that images can be subtracted meaningfully. In this way, the evolution of a tumor can be monitored through the comparison of two images produced by the same imaging modality for the same patient at different times, or an abnormality can be classified by comparing it to the reference images in an anatomical atlas (hence, computer-aided diagnosis). It is necessary that these registration operations are done with subpixel accuracy for diagnosis reliability. Especially if any post-processing will be done, such as automatic detection of tumors, the success of this detection depends on the accuracy of the registration. Data comparison is also frequently used in the field of remotely sensed data processing. Surveillance of changes in lands, nuclear plants, natural sources or monitoring the growth of urban areas are some examples [9, 10]. Aligning the images of an area with a reference map, target recognition, or finding a match for a reference pattern in an image for the purpose of classification of well-defined scenes such as airports and roads, are common applications of data comparison [11, 12]. Subpixel registration is especially important in the alignment of aerial images. For example, a one-pixel distance for a Landsat image corresponds to a distance of about 80 meters on the Earth, so that pixel-level registration provides only  $\pm 40$ -meter resolution [13]. In order to achieve a  $\pm 4$ -meter resolution on the Earth, 0.1 pixel accuracy must be achieved by the registration. Other interesting applications of data comparison are character recognition [14] and signature verification.

Subpixel image registration is also employed in image fusion. The goal of image fusion is to integrate complementary information from images obtained

by different sensors such that the fused image is more suitable for the purpose of human visual perception and computer-processing tasks such as segmentation, feature extraction, and object recognition [15]. A common application of image fusion is in medical image analysis. For example, it is desirable to combine multi-modality medical images that contain structural information (MRI, CT) with images that contain functional information (PET, SPECT) for better diagnosis [15, 16, 17, 18, 19, 20, 21]. In this way, structural and functional information for the same region of the body can be localized better. Another example of data fusion is from the field of remotely sensed data processing. Here, data fusion is used to integrate images from different electromagnetic bands [22] (e.g., microwave, radar, infrared, and visual) for classification of buildings, roads, vehicles, etc.

Subpixel image registration may also be employed for noise reduction purposes [23]. The availability of multiple instances of the same data is advantageous in this case. For example, by simple averaging, or by using more complex processes [24], the registration of these instances makes the extraction of common features possible [23]. This approach has been successfully applied to the correlation-averaging of virus particles in high-resolution electron microscopy [23].

Two other applications of subpixel image registration are in the areas of deinterlacing and resolution improvement [25]. The former one refers to obtaining a still picture from an interlaced video, for viewing on a progressive monitor or printing. Resolution improvement, on the other hand, refers to up-converting the spatial sampling grid used by a given image sensor to produce the effect of a zoom. The methods used for resolution improvement involve exploiting the temporal redundancies that exist within digital video signal by combining some form of interpolation with subpixel displacement estimation [25].

## 1.2 Literature Review

The existing image registration techniques can be broadly classified as search techniques, correlation techniques, differentiation techniques, and feature

matching techniques and others. In the following, we give an overview of literature in each of these classes.

### 1.2.1 Search Techniques

The common approach used by the search techniques is as follows. First, a feature space is selected which determines the features of images that will be matched. The most common features of images used in the literature are raw pixel intensities, edges [9, 26], contours, surfaces, corners, line intersections, and moment invariants [27, 18]. Then, the search space determines the set of transformations that is capable of registering the images. These transformations can include translation, rotation, zoom, shear, or more complex motion. Transformations used to align two images can be applied globally or locally. A global transformation is composed of a single relation that is the same at each location of the image. In local transformations, on the other hand, the transformation parameters may change depending on the spatial locations of pixels in the images. These global (local) transformations are used to correct the global (local) variations between two images. In certain cases, we may not want to eliminate all the variations between the images because some of the variations could be differences to be detected after the registration. In such cases, a global transformation might be chosen even though the images contain local variations.

Once a feature space and a search space are selected, a search strategy is employed to determine the rules of selecting suitable transformations from the search space. Commonly used search strategies include dynamic programming [28], linear programming [1], simulated annealing [4], genetic algorithms [4], adaptive random search [16], and simplex method [16]. The most commonly used search strategies for determining the transformation parameters in the case of translational motion, however, are the exhaustive, logarithmic, and cross search [29] methods. Among these methods, the logarithmic search is the most popular one because it offers a compromise between the search accuracy and the size of the search space. The logarithmic search method is described in Appendix B.2.

In order to reduce the computational load of the search-based algorithms, attempts have been made to reduce the space to be searched [4]. This is done by adaptively estimating an optimal subspace from the maximum allowable space during the search process. The optimal subspace consists of a fraction of the best structures that have been encountered by the search at a certain iteration. The chosen search strategy is employed until a similarity metric gives a satisfactory or the best result. Some similarity metrics can be listed as the sum of absolute differences of intensity [1, 30], sum of squared differences of intensity, sum of absolute differences of contours, surface differences, number of matching bits between the corresponding pixels [5, 31], number of sign changes in pointwise intensity difference [7], and histogram of pixel intensities of the difference image [17, 6].

### 1.2.2 Correlation Techniques

Maximizing the cross-correlation between two images by a proper selection of a transformation is a basic approach in image registration [3, 32, 1, 33]. Cross-correlation is a similarity metric that measures the degree of similarity between two images. The cross-correlation method is useful for small rigid and affine transformations. The location of the peak of the cross-correlation function gives the actual displacement between the images. The cross-correlation method can provide subpixel accurate results [13] by employing a cross-correlation surface interpolation approach.

There is also a class of correlation techniques that are based on the Fourier transform. They utilize the nice properties of the Fourier transform due to translation, rotation, and scale changes of the images in the spatial domain. Among these techniques is the celebrated phase-correlation method, which is first proposed by Kuglin and Hines in [3] for aligning two images which are spatially shifted with respect to each other. The phase-correlation method is based on detecting the location of the peak of the inverse Fourier transform of the normalized cross-power spectrum of the images to be aligned (Appendix A). It is much more easier to detect the peak of the phase correlation function (which

is ideally a delta function) than to detect the peak of the cross-correlation function [3]. The phase correlation method is also robust to intensity (magnitude only) variations between images because of the normalization of the cross-power spectrum. It is also insensitive to convolutional image degradations. Since all spectral phase terms are treated equally due to the “whitening” effect of the normalization of the cross-power spectrum, phase correlation algorithm is robust in the presence of narrow bandwidth noise. One drawback of the phase-correlation algorithm is that it is more sensitive to noise than direct cross-correlation is [33]. As described in [3, 32] the amplitude of the peak is a direct measure of the degree of congruence between the two aligned images. The phase correlation SNR ratio can be expressed as a function of the peak amplitude and the square root of the total number of sample points [3].

The phase-correlation method can also give subpixel accurate results by employing phase-correlation surface interpolation approach [3, 32, 1]. This correlation surface interpolation approach to subpixel displacement estimation is described in detail in Appendix B.1. It is also possible to detect multiple moving objects in a scene using the phase correlation method. This is done by detecting multiple peaks in the phase correlation function, each representing the motion of a different object [29].

A filtered phase correlation technique is also presented in [3, 32] by introducing a multiplicative weighting function to the normalized cross-power spectrum of the images to be registered. In this way, a filtered phase correlation function is obtained. The resulting family of filtered phase correlation algorithms include both phase correlation and cross-correlation algorithms.

Several methods are proposed in the literature to reduce the computational load of the phase correlation algorithm by using projections [34, 35]. The method given in [34] employs the projection-slice theorem and uses the 1-D Fourier transform in the phase correlation algorithm instead of the 2-D Fourier transforms.

An extension of the phase correlation method which covers both translational and rotational movements is introduced in [36]. It is assumed that one

of the images is a rotated and translated replica of the other image. The proposed method is to search over possible rotation angles and to choose the one that gives the maximum peak in the phase-correlation surface.

Another extension of the phase correlation method corrects for translational, rotational, and scale changes [37]. The method uses Fourier scaling properties and Fourier rotational properties in polar coordinates with a logarithmic scaling, since scaling and rotation of an image in the spatial domain correspond to translational shifts of its Fourier transform in the polar coordinates. The algorithm first estimates the amounts of rotation and scaling, and then estimates the amount of displacement, both by employing the phase-correlation algorithm.

Finally, power cepstrum and spectrum functions are used in [38] for determining the rotational and translational misregistration parameters. This technique employs the idea of changing the rotational shift into a translational shift by using the shift-invariant property of the power spectrum.

### 1.2.3 Differentiation Techniques

Differentiation techniques employ intra-frame and inter-frame image gradients for the estimation of the motion vector between two images, or blocks of images [29, 13, 39]. In particular, the Lucas-and-Kanade method [29] is a well known differentiation-based technique which is derived from the optical flow equation. In this method, motion is modeled by a simple translation and the two displacement parameters are obtained with subpixel accuracy. One drawback of this method is that it is not robust to intensity variations. (This fact will be shown using the simulation results in Chapter 4.) In general, optical-flow based differentiation techniques assume that a complex moving scene will be indistinguishable from a single pattern undergoing simple translation when viewed through a sufficiently small window over a sufficiently short interval of time [40]. An extension of the translational motion model to the affine motion is given in [40]. The parameters of the affine transformation are obtained by solving a linear system of six equations involving image gradients in their coefficients.

Another differentiation-based displacement estimation method is the Horn-and-Schunk method [29]. This method aims to find a motion field that satisfies the optical flow equation with minimum pixel-to-pixel variations among the displacement vectors. The method given in [39] also uses the image gradients as in the Horn and Schunk method, but employs a polynomial transformation model.

#### 1.2.4 Feature Matching Techniques and Others

When the misregistration type between two images is unknown, the *feature or landmark-matching approach* is used [1, 41]. The general approach for feature mapping algorithms goes as follows. First, the features or control points of the images are extracted. It is desirable that this extraction process is automatic rather than manual. Then, a correspondence between the control points of both images is established. Finally, a spatial mapping which usually consists of two 2-D polynomial transformations (one for each coordinate in the registered image) is determined using the matched control points [1].

Control points that are selected for matching can be either intrinsic or extrinsic [1]. Intrinsic control points are not related with the data and placed in the scene intentionally for registration purposes. In medical imaging, for example, fiducial marks and head-holders are placed in known positions in the patients during imaging, to act as a reference [19]. Of course, this is not a comfortable process for the patients and even more, it can not eliminate some autonomous motions like beating of the heart and breathing. Such devices are also difficult to use across modalities like CT and MRI [42]. Extrinsic control points [11] are derived from the data, either manually or automatically. Some typical features used as control points are corners, line intersections, contour points [8], centers of windows having locally maximum curvature, and centers of gravity of closed-boundary regions. Control points should be chosen such that they are likely to be uniquely found in both images.

There are many methods for matching the selected control points [14, 10], which include relaxation [43], clustering methods, least squares method [14], and accumulator algorithm [10]. After the transformation that matches the

control points has been estimated, the function that matches the whole images can be chosen either as being global or local [14]. Some local transformations use 2-D mesh techniques by triangulation of control points [44]. After the triangulation, a linear mapping function is obtained by registering each pair of corresponding triangular regions in the images. Morphological transformations based on matching the edges from both images are also reported in the literature [26, 9].

Other types of image registration techniques include segmentation-based methods, which have been shown to be more robust to noise than the point-based registration methods [10]. There are also approaches which are based on the principal axis transformation [42]. Several recently developed methods employ the wavelet coefficients of the images for registration with an application to the fusion of multi-focus and aerial images [15]. This technique appropriately combines the wavelet transforms of the input images, and the fused image is obtained by taking the inverse wavelet transform of the fused wavelet coefficients.

### 1.3 Contribution and Scope

A contribution that has already been made in this thesis is the literature review of existing image registration techniques. Comparison of the performances of existing subpixel displacement methods is another contribution of this thesis. In the following chapters, the comparison results, which are obtained under different noise levels, based on both real and synthetically generated sequences are presented.

The main contribution of the thesis is the introduction of a novel closed-form solution which registers images with subpixel accuracy. It is assumed that the misregistration is due to global translational motion. Given the pixel part of the displacement, the proposed method estimates the subpixel part of the displacement between two frames of an image sequence. The pixel part of the displacement is found using the phase correlation technique. The similarity

criterion used is the mean-squared error over the displaced frames. Image intensities at subpixel locations are evaluated using bilinear interpolation.

The proposed method is designed so that it is insensitive to frame-to-frame intensity variations. It is both faster and more accurate than the existing search-based solutions. It also outperforms other closed-form solutions, such as the differentiation and correlation surface fitting based techniques, in terms of both accuracy and computational complexity.

The main application of the proposed closed-form solution is in correction of unsteadiness in digitized motion pictures. It is also applied to the de-interlacing problem. Even and odd fields of an interlaced video frame may have translational shifts with respect to each other due to relative movements of the camera or the scene during recording. When even and odd fields of such a frame are displayed simultaneously to obtain a still image, the quality of the image is usually quite poor. This problem is properly eliminated by registering even and odd fields with subpixel accuracy using the near-closed-form solution. An interlaced unstable sequence is also stabilized using the near-closed-form solution by registering even fields of consecutive frames. This approach eliminates the need to convert the interlaced sequence to progressive format.

Chapter 2 introduces a new search-based approach that is more efficient than exhaustive and logarithmic search methods. A comparison of the three search methods based on their speed and accuracy is also provided.

Chapter 3 gives the derivations of the near-closed-form solution when there is no intensity variations between consecutive frames of a sequence. Simulation results are also given for synthetically generated Text and CT Sequences and a real Bilkent Sequence. The algorithm is also tested for three different noise levels using the synthetic Text Sequence. An application of the proposed algorithm to de-interlacing is also presented.

Chapter 4 extends the near-closed-form approach to sequences that contain contrast and brightness changes. Simulations which examine different illumination models are also presented.

Chapter 5 gives the conclusions and future work.

'

## Chapter 2

# AN EFFICIENT SEARCH METHOD

### 2.1 Background

Let  $s_1(.,.)$  denote the reference frame and  $s_2(.,.)$  denote the misregistered frame after having been corrected for any integer pixel displacement by using the phase correlation method (described in Appendix A). Then,  $s_1()$  and  $s_2()$  differ from each other only by a subpixel displacement  $(d_1, d_2)$  (assuming that the only cause of misregistration is a displacement), i.e.,

$$s_1(n_1, n_2) = s_2(n_1 + d_1, n_2 + d_2), \quad -1 < d_1, d_2 < 1. \quad (2.1)$$

We employ bilinear interpolation to approximate the value of  $s_2(n_1 + d_1, n_2 + d_2)$ . That is, for positive  $d_1$  and  $d_2$

$$\begin{aligned} \tilde{s}_2(n_1 + d_1, n_2 + d_2) &= s_2(n_1, n_2)(1 - d_1)(1 - d_2) \\ &\quad + s_2(n_1 + 1, n_2)(d_1)(1 - d_2) \\ &\quad + s_2(n_1, n_2 + 1)(1 - d_1)(d_2) \\ &\quad + s_2(n_1 + 1, n_2 + 1)(d_1)(d_2). \end{aligned} \quad (2.2)$$

which is also illustrated in Figure 2.1.

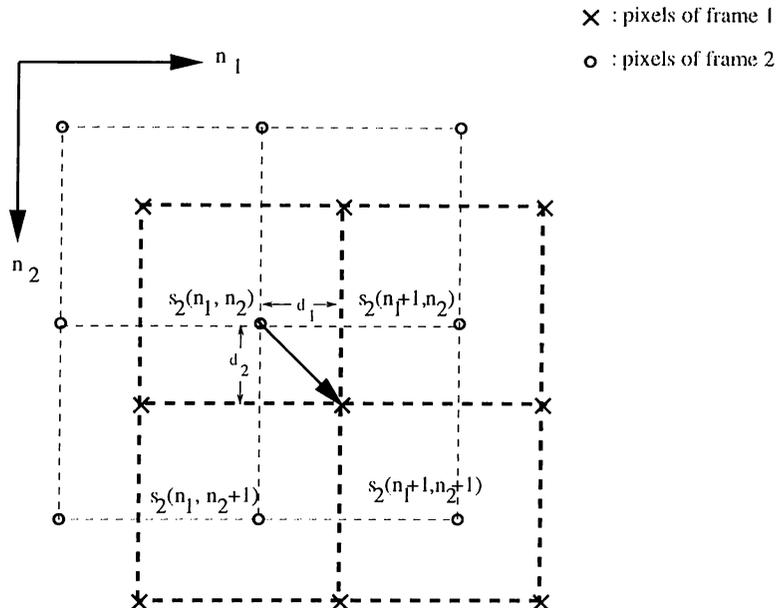


Figure 2.1: The bilinear interpolation for positive  $d_1, d_2$ .

We define a mean-squared error function

$$\text{MSE} = \frac{1}{N_1 N_2} \sum_{n_1, n_2 \in \mathcal{B}} [s_1(n_1, n_2) - \tilde{s}_2(n_1 + d_1, n_2 + d_2)]^2, \quad d_1, d_2 \geq 0, \quad (2.3)$$

where  $\mathcal{B}$  denotes an  $N_1 \times N_2$  block of pixels over which the MSE is computed. The problem of estimating the subpixel displacement can now be stated as finding  $d_1, d_2$  that minimizes MSE such that  $0 \leq d_1, d_2 < 1$ .

A straightforward approach to minimizing (2.3) would be to uniformly sample the set  $\{(d_1, d_2) : 0 \leq d_1, d_2 \leq 1\}$  at a desired accuracy, compute the MSE given in (2.3) for every sample pair  $(d_1, d_2)$ , and pick the pair that minimizes the MSE.

In exhaustive (full) search, all possible locations up to the desired accuracy are tested and the subpixel displacement which minimizes the MSE is chosen. Figure 2.2 explains this procedure, where crosses denote actual pixel locations. For example, if an accuracy of  $1/4$  pixels is desired, all the subpixel displacements at the intersections of the lines in Figure 2.2 should be evaluated. This is done by shifting the frame to be registered by the subpixel displacement

at hand, using bilinear interpolation. If an accuracy of  $2^{-n}$  pixels is desired, the exhaustive search requires the evaluation of (2.3) for  $(2^{n+1} - 1)^2$  different values of  $d_1$  and  $d_2$ . This corresponds to a total of  $N_1 N_2$  bilinear interpolations which results in  $9N_1 N_2 (2^{n+1} - 1)^2$  multiplications and  $5N_1 N_2 (2^{n+1} - 1)^2$  summations. Since  $n$  appears as the power in these expressions, the number of multiplications and summations increase by approximately 16 times when  $n$  is doubled. This brings a large computational load which can be significantly reduced by using the logarithmic search technique (described in Appendix B.2). However, both exhaustive and logarithmic search techniques are quite time consuming, because for each new  $(d_1, d_2)$ , bilinear interpolation for shifting one of the frames is carried out from the beginning.

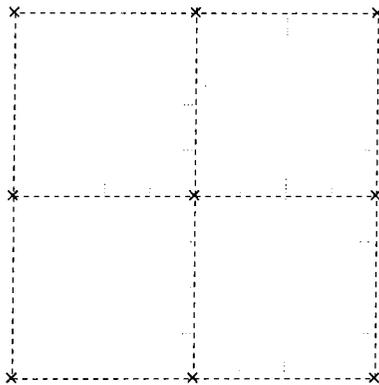


Figure 2.2: Exhaustive search

The search method proposed in this chapter is more efficient than the exhaustive and logarithmic search methods because the bilinear interpolation is employed only once in the formulation. We present this computationally more efficient search-based solution in the next section.

## 2.2 Method

We rewrite (2.2) for all  $-1 < d_1, d_2 < 1$ , as

$$\tilde{s}_2(n_1 + d_1, n_2 + d_2) = S_0^{(i)} + S_1^{(i)} d_1 + S_2^{(i)} d_2 + S_3^{(i)} d_1 d_2, \quad (d_1, d_2) \in \mathcal{Q}^{(i)}, \quad (2.4)$$

where  $\mathcal{Q}^{(i)}$ ,  $i = 1, 2, 3, 4$ , denote the four quadrants defined as

$$\begin{aligned}\mathcal{Q}^{(1)} &= \{(d_1, d_2) : 0 \leq d_1, d_2 < 1\}, \\ \mathcal{Q}^{(2)} &= \{(d_1, d_2) : 0 \leq d_1 < 1, -1 < d_2 < 0\}, \\ \mathcal{Q}^{(3)} &= \{(d_1, d_2) : -1 < d_1 < 0, 0 \leq d_2 < 1\}, \\ \mathcal{Q}^{(4)} &= \{(d_1, d_2) : -1 < d_1, d_2 < 0\},\end{aligned}\tag{2.5}$$

and the coefficients  $S_0^{(i)}, S_1^{(i)}, S_2^{(i)}, S_3^{(i)}$  are functions of the intensities at pixels neighboring to  $(n_1, n_2)$ ; they are defined as

$$\begin{aligned}S_0^{(i)} &= s_2(n_1, n_2) \\ S_1^{(i)} &= I[s_2(n_1 + I, n_2) - s_2(n_1, n_2)] \\ S_2^{(i)} &= J[s_2(n_1, n_2 + J) - s_2(n_1, n_2)] \\ S_3^{(i)} &= IJ[s_2(n_1 + I, n_2 + J) - s_2(n_1 + I, n_2) - s_2(n_1, n_2 + J) + s_2(n_1, n_2)],\end{aligned}\tag{2.6}$$

where

$$I = \begin{cases} 1 & \text{for } i = 1, 2 \\ -1 & \text{for } i = 3, 4 \end{cases}, \quad \text{and} \quad J = \begin{cases} 1 & \text{for } i = 1, 3 \\ -1 & \text{for } i = 2, 4 \end{cases}\tag{2.7}$$

We rewrite (2.3) for all  $-1 \leq d_1, d_2 \leq 1$ , as

$$\text{MSE}^{(i)} = \frac{1}{N_1 N_2} \sum_{n_1, n_2 \in \mathcal{B}} [s_1(n_1, n_2) - \tilde{s}_2(n_1 + d_1, n_2 + d_2)]^2, \quad (d_1, d_2) \in \mathcal{Q}^{(i)}.\tag{2.8}$$

The problem of estimating the subpixel displacement can now be stated as finding  $(d_1, d_2)$  in  $\mathcal{Q}^{(i)}$  that minimizes  $\text{MSE}^{(i)}$  for each  $i = 1, 2, 3, 4$ . Then, we pick the pair  $(d_1, d_2)$  that results in the overall minimum MSE.

From (2.8) and (2.4) we obtain the following expression for  $\text{MSE}^{(i)}$  in terms of the subpixel shifts  $d_1$  and  $d_2$ :

$$\begin{aligned}\text{MSE}^{(i)} &= C_0^{(i)} + C_1^{(i)}d_1 + C_2^{(i)}d_2 + C_3^{(i)}d_1d_2 + C_4^{(i)}d_1^2 + C_5^{(i)}d_2^2 \\ &\quad + C_6^{(i)}d_1^2d_2 + C_7^{(i)}d_1d_2^2 + C_8^{(i)}d_1^2d_2^2, \quad i = 1, 2, 3, 4,\end{aligned}\tag{2.9}$$

where the coefficients  $C_0^{(i)}, \dots, C_8^{(i)}$  are computed over the two images using the basic summations as described in Appendix C.1 and Appendix C.2. The

computation of the coefficients in (2.9) require at most a total of 39 summations over a block of pixels in the two images which results in approximately  $(39N_1N_2)$  multiplications and  $(39N_1N_2)$  summations.

Once the coefficients  $C_0^{(i)}, \dots, C_8^{(i)}$  have been computed for all the four quadrants, that is for  $\mathcal{Q}^{(1)}, \dots, \mathcal{Q}^{(4)}$ , search over possible  $(d_1, d_2)$  values up to the desired subpixel accuracy can be carried out by inserting current  $d_1$  and  $d_2$  values into (2.9). The search strategy can be chosen as exhaustive or logarithmic. If exhaustive search strategy is chosen, the method requires a total of  $(39N_1N_2) + 14(2^{n+1} - 1)^2$  multiplications and  $(39N_1N_2) + 14(2^{n+1} - 1)^2$  summations. Note that 14, the coefficient of  $(2^{n+1} - 1)^2$ , is much smaller than  $9N_1N_2$  which is required for traditional exhaustive search.

Thus the steps of the proposed algorithm can be summarized as follows:

1. Compute the basic summations in Appendix C.1 over a specified block of pixels.
2. Compute the MSE coefficients  $C_0^{(i)}, \dots, C_8^{(i)}$  as given in Appendix C.2 for each quadrant.
3. Using the coefficients  $C_0^{(i)}, \dots, C_8^{(i)}$  and the MSE expression (2.9), search (logarithmically or exhaustively) over the set of possible  $(d_1, d_2) \in \mathcal{Q}^{(i)}$  values up to the desired accuracy and pick the one that minimizes the  $\text{MSE}^{(i)}$  given in (2.9). This displacement is called as the candidate of quadrant  $\mathcal{Q}^{(i)}$ .
4. Among the four candidate displacements for each quadrant, pick the one with the minimum MSE.

## 2.3 Results

The methods are tested on a synthetically generated “Text” sequence. This sequence is generated as follows. First, a large synthetic Text image which has dimensions  $697 \times 356$  is created. The background has a grey level of 60 (white

is 255) and the text has a gray level of 200. A  $5 \times 5$  uniform blur is applied to this image to smooth out the edges of the letters. In order to simulate a sequence with subpixel displacements, first, frames with displacements of 0,  $\pm 1$ ,  $\pm 2$ ,  $\pm 3$ , and  $\pm 4$  pixels with respect to the reference frame are generated in a random order. Then, these frames are spatially down sampled by a factor of 5, so that they have random subpixel displacements of 0,  $\pm 0.2$ ,  $\pm 0.4$ ,  $\pm 0.6$ , and  $\pm 0.8$  pixels with respect to the reference frame. Finally, to simulate the observation noise, a 10 dB zero-mean white Gaussian noise is added to each frame. This means that the standard deviation  $\sigma_n$  of the noise is adjusted so that the Peak-Signal-to-Noise Ratio (PSNR) is 10 dB, i.e.,

$$\text{PSNR} = 10 \log \frac{255}{\sigma_n^2} = 10dB. \quad (2.10)$$

The reference frame of the Text sequence is shown in Figure 2.3. Using the above procedure, 20 frames are generated. When these frames are displayed sequentially at the rate of 30 frames/second, the subpixel displacements indeed cause a disturbing jitter. The random displacements that are chosen for these frames are given in Table 2.1. The same random displacements will be used throughout the thesis.

The evaluation of (2.9) for a given  $(d_1, d_2)$  has taken insignificant CPU time ( $< 16$  msec.) on SunSparc20. The only time consuming part of this search approach is finding the coefficients  $C_0^{(i)}, \dots, C_8^{(i)}$ .



Figure 2.3: The first frame of the Text sequences.

In order to register the Text sequence, displacements of each frame is estimated with respect to the first, i.e., the reference frame. A  $7 \times 7$  uniform blur is applied to each frame prior to subpixel displacement estimation to reduce the effects of bilinear interpolation and any additive noise. The simulations using the Text sequence are carried out for three search methods, namely the

Frame No.	$d_1$	$d_2$
2	0.2	-0.8
3	-0.2	-0.4
4	0.8	0.6
5	-0.8	0.2
6	0.4	0.4
7	0.2	0.6
8	0.8	0.6
9	-0.8	0.4
10	-0.8	-0.6
11	-0.8	0.6
12	-0.2	0.8
13	-0.2	-0.6
14	0.6	-0.4
15	0.8	0.2
16	-0.2	0.8
17	0.6	0.4
18	0.8	-0.8
19	0	0.2
20	0.8	0.8

Table 2.1: The random displacements of the frames in the Text sequence. The first frame is the reference frame and has frame number 1.  $d_1$  and  $d_2$  denote the vertical and horizontal displacement, respectively.

proposed, exhaustive, and logarithmic search methods using two different sub-pixel accuracies, 1/8 and 1/16 pixels, and two different block sizes,  $100 \times 100$  and  $50 \times 50$  blocks.

In Table 2.2, mean of absolute displacement error values for  $d_1$  and  $d_2$  are compared for the three methods. All three algorithms give the same results for the same search accuracy irrespective of the block size. Because the values of  $d_1$  and  $d_2$  are searched exhaustively in the proposed solution, the proposed search and the conventional solutions are indeed expected to give the same results. The results also show that although the logarithmic search is suboptimal, it is capable of performing as well as the exhaustive search.

The three techniques, on the other hand, show significant differences when compared in terms of the computational times required by each one of them. Table 2.3 gives the CPU times (on a SunSparc20) of the three search methods for different block sizes and search accuracies. It can easily be seen that the proposed method is by far the fastest and the conventional exhaustive search is the slowest among the three. Hence the superiority of the proposed search method over the widely used exhaustive and logarithmic search methods. It is also equally important to note that, in the proposed search method, the CPU time does not increase significantly with the accuracy of the solution unlike other methods. The search time for the proposed method can be further reduced by using a smaller-sized block of pixels for the computation of the coefficients  $C_0^{(i)}, \dots, C_8^{(i)}$ .

	1/8 pixel accuracy	1/16 pixel accuracy
$\epsilon(d_1)$	0.0434	0.0151
$\sigma(d_1)$	0.0140	0.0089
$\epsilon(d_2)$	0.0368	0.0224
$\sigma(d_2)$	0.0153	0.0098

Table 2.2: The mean of absolute displacement errors ( $\epsilon(d_1)$  and  $\epsilon(d_2)$ ) and the standard deviation of absolute displacement errors ( $\sigma(d_1)$  and  $\sigma(d_2)$ ) found by the proposed search, exhaustive search and logarithmic search methods for 1/8 and 1/16 pixel search accuracies.

	Proposed method	Logarithmic search	Exhaustive search
Case 1	0.0259	0.3438	3.6670
Case 2	0.0498	0.6605	7.0650
Case 3	0.0340	0.4421	13.400
Case 4	0.0654	0.8956	27.470

Table 2.3: CPU times (in seconds) for the three search-based methods. Case 1: A: 1/8 pixels, B:(50 × 50), Case 2: A: 1/8 pixels, B:(100 × 50), Case 3: A: 1/16 pixels, B:(50 × 50), and Case 4: A: 1/16 pixels, B:(100 × 50) where A denotes the accuracy and B denotes the block size.

## Chapter 3

# A NEAR-CLOSED-FORM SOLUTION

In this chapter, a novel near-closed-form solution to subpixel displacement estimation is introduced. Thus, the proposed solution is not based on a search-based technique. A comparison of the method with the popular subpixel displacement estimation methods, namely, logarithmic search, phase-correlation surface interpolation, differentiation and cross-correlation surface interpolation are given using synthetic and real sequences. The experiments demonstrate the superiority of the proposed near-closed-form solution. Application of the near-closed-form solution to de-interlacing, as well as unsteadiness correction, is also presented.

### 3.1 Method

We repeat for convenience the following expression for the MSE that was originally derived in Chapter 2

9

$$\begin{aligned} \text{MSE}^{(i)} = & C_0^{(i)} + C_1^{(i)}d_1 + C_2^{(i)}d_2 + C_3^{(i)}d_1d_2 + C_4^{(i)}d_1^2 + C_5^{(i)}d_2^2 \\ & + C_6^{(i)}d_1^2d_2 + C_7^{(i)}d_1d_2^2 + C_8^{(i)}d_1^2d_2^2, \quad i = 1, 2, 3, 4, \end{aligned} \quad (3.1)$$

where  $(i)$  denotes one of the four quadrants in the cartesian coordinates as defined in Chapter 2. In this chapter, we are after the analytical minimization of  $\text{MSE}^{(i)}$  rather than a search-based minimization.

In order to minimize  $\text{MSE}^{(i)}$  with respect to  $d_1$  and  $d_2$ , we solve  $\partial\text{MSE}^{(i)}/\partial d_1 = 0$  and  $\partial\text{MSE}^{(i)}/\partial d_2 = 0$  simultaneously:

$$\frac{\partial\text{MSE}^{(i)}}{\partial d_1} = C_1^{(i)} + C_3^{(i)}d_2 + 2C_4^{(i)}d_1 + 2C_6^{(i)}d_1d_2 + C_7^{(i)}d_2^2 + 2C_8^{(i)}d_1d_2^2 = 0 \quad (3.2)$$

$$\frac{\partial\text{MSE}^{(i)}}{\partial d_2} = C_2^{(i)} + C_3^{(i)}d_1 + 2C_5^{(i)}d_2 + C_6^{(i)}d_1^2 + 2C_7^{(i)}d_1d_2 + 2C_8^{(i)}d_1^2d_2 = 0. \quad (3.3)$$

We note that the equation (3.2) is linear in  $d_1$ . Thus we can express  $d_1$  as a function of  $d_2$  as

$$d_1 = -0.5 \frac{C_1^{(i)} + C_3^{(i)}d_2 + C_7^{(i)}d_2^2}{C_4^{(i)} + C_6^{(i)}d_2 + C_8^{(i)}d_2^2}. \quad (3.4)$$

Then, we substitute (3.4) in the equation (3.3), to obtain the following polynomial equation in  $d_2$

$$E_5d_2^5 + E_4d_2^4 + E_3d_2^3 + E_2d_2^2 + E_1d_2 + E_0 = 0, \quad (3.5)$$

where the coefficients  $E_0, \dots, E_5$ , are defined in terms of  $C_0, \dots, C_8$ . The definitions of  $E_0, \dots, E_5$  are given in Appendix C.3. Unfortunately, there does not exist an algebraic formula for the zeros of a fifth degree polynomial. Thus, the zeros of (3.5) are obtained numerically using the Muller's method [45]. Once the solution for  $d_2$  is obtained,  $d_1$  is calculated from (3.4).

Because (3.5) is a fifth degree polynomial, for each quadrant  $\mathcal{Q}^{(i)}$ , at least one of the roots will be real and the remaining two pairs may be complex conjugates of each other. Among the roots obtained for quadrant  $\mathcal{Q}^{(i)}$ , only the solutions  $(d_1, d_2)$  that are in  $\mathcal{Q}^{(i)}$  are accepted. In the case there is more than one acceptable solution for  $(d_1, d_2)$  considering all quadrants, the solution with

the minimum MSE is picked to be the actual subpixel displacement. On the other hand, when there is no acceptable solution at all—this actually happened very rarely in our experiments, the proposed algorithm defaults to the efficient search method proposed in Chapter 2 to find the subpixel displacement. Hence the name *near-closed-form* solution.

Thus the steps of the proposed algorithm can then be summarized as follows:

1. Compute the basic summations  $A_{0,0}$ ,  $A_{0,0;0,0}$ ,  $B_{0,0;i,j}$ ,  $D_{i,j}$ ,  $D_{i,j;k,\ell}$ , given in Appendix C.1 over a specified block of pixels. Note that only 39 basic summations are computed at this step.
2. Compute the MSE coefficients  $C_0^{(i)}, \dots, C_8^{(i)}$ , given in Appendix C.2 for each quadrant, i.e., for each  $i = 1, 2, 3, 4$ .
3. Compute the coefficients  $E_0, \dots, E_5$ , of the fifth degree polynomial as given in Appendix C.3 for each quadrant.
4. Find the zeroes of (3.5) for each quadrant. Among the acceptable ones, pick the one with the minimum MSE. That gives the near-closed-form solution. If there is no solution, find  $(d_1, d_2)$  which minimizes the MSE expression (3.1) using the efficient search method of Chapter 2.

## 3.2 Results

The simulations in this chapter are carried out on three different sequences, namely the Text Sequence, the CT sequence, and the Bilkent Sequence. While the first two sequences are synthetically generated, the third sequence is obtained from an actual video recording. In the following, over these sequences, the performance of the proposed near-closed-form solution is compared with the commonly used subpixel displacement estimation methods existing in the literature. The methods that the near-closed-form solution is compared against are, the logarithmic search (Appendix B.2), differentiation method (Appendix B.3), phase-correlation and cross-correlation surface interpolation

methods (Appendix B.1). We note that a  $7 \times 7$  uniform blur is applied to all images in Text Sequences prior to displacement estimation in order to reduce the effects of bilinear interpolation and any additive noise. The blur is chosen as  $5 \times 5$  for the Bilkent and CT sequences since they contain less noise. The accuracy of the logarithmic search is chosen as  $1/16$ , i.e., 0.0625, pixels.

### 3.2.1 Simulations with the synthetic Text Sequence

This section presents the simulation results using the synthetically generated Text Sequence that consists of 20 frames. The generation of this sequence is described in detail in Section 2.3. The first frame of the Text Sequence with 5 dB PSNR is shown in Figure 3.1.

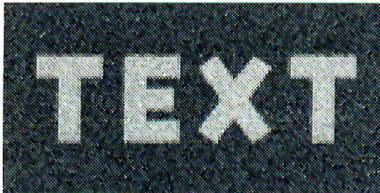


Figure 3.1: The first frame of the Text Sequence for 5 dB SNR.

In the following, we present the results obtained with the subpixel displacement estimation methods at three different noise levels, namely, at PSNR's of 20 dB, 10 dB and 5 dB. The plots in Figures 3.2, 3.3, and 3.4 compare the absolute displacement estimation errors of the near-closed-form solution for each frame with those of the other methods, at 20 dB, 10 dB, and 5 dB, respectively. In order to see these results compactly, Table 3.1 is provided, which shows the mean of absolute errors for 19 frames of the Text Sequence (the first frame, which is the reference frame, is not shown here) for the three noise levels.

As it can be seen from Figures 3.2, 3.3, and 3.4, the absolute displacement error for the near closed-form solution is smaller than that of the logarithmic search and the phase correlation algorithm for almost all of the frames, at all three noise levels. In fact, when we compare the mean absolute displacement estimation errors given in Table 3.1, we observe that the performance of the near closed-form solution is significantly superior to that of the logarithmic

search and phase correlation at all three noise levels. When compared to the differentiation and cross correlation algorithms, on the other hand, the near-closed-form solution may result in larger absolute displacement errors for some of the frames. However, as seen from Table 3.1, on the average, the performance of the near closed-form solution is still better than the differentiation and cross correlation algorithms at both 20 dB and 10 dB noise levels. At the 5 dB noise level, while the near closed form solution still performs better than the differentiation method, its performance is about the same as that of the cross correlation method. As far as the CPU times (Table 3.1) are concerned, we can say that all methods require about the same computational time except for the logarithmic search method, which is considerably slower than the rest of the methods.

### 3.2.2 Simulations with the CT Sequence

The CT Sequence is generated from a real computed tomography (CT) image, shown in Figure 3.5, that has dimensions of  $512 \times 512$ . The frames of this sequence are generated in the same way as described for the Text Sequence in Section 2.3. Thus, there are 20 frames in the CT Sequence with subpixel displacements of 0,  $\pm 0.2$ ,  $\pm 0.4$ ,  $\pm 0.6$ , and  $\pm 0.8$  pixels with respect to the reference frame. These frames are of size  $102 \times 102$  due to the 5-to-1 subsampling of the original  $512 \times 512$  CT image to simulate the subpixel displacements. The CT image shown in Figure 3.5 belongs to an axial cross-section of the body around the liver. This CT Sequence simulates in a sense, the CT images of the same patient, from the same slice of the body taken at different times. By registering these slices it will be easier to detect any changes that could be caused by tumors.

A  $60 \times 60$  block of pixels is chosen, centered around the backbone, for the calculation of the summations  $A_{0,0}$ ,  $A_{0,0; 0,0}$ ,  $B_{0,0; i,j}$ ,  $D_{i,j}$ ,  $D_{i,j; k,\ell}$ , given in Appendix C.1. In general, a block with sufficient gray level variations should be chosen for the subpixel displacement estimation. Figure 3.6 gives the comparison of absolute displacement estimation errors of the near-closed-form solution with other methods.

As it can be seen from Figure 3.6, in the case of the CT Sequence, the absolute displacement error for the near closed-form solution is smaller than that of the phase correlation and cross correlation algorithms for almost all of the frames. When compared to the logarithmic search and differentiation algorithms, the near-closed-form solution may result in larger absolute displacement errors for some of the frames. However, as seen from Table 3.2, on the average, the performance of the near closed-form solution is still better than the logarithmic search and differentiation methods. The CPU times (Table 3.2) used by the five methods compare as in the previous section, i.e., all methods require about the same computational time except for the logarithmic search method, which in this case is about two times slower than the rest of the methods.

### 3.2.3 Simulations with the Bilkent Sequence

In this section, we present simulation results carried on a real colored sequence with unknown displacements. This sequence is named as the Bilkent Sequence since it contains the Bilkent emblem. The stationary Bilkent emblem is recorded using a hand-held camcorder (SONY Video-8) over a few seconds on a Betacam video cassette. Unknown displacements occur between the frames (and fields) of the video due to the movements of the hand of the person holding the camcorder during recording. These displacements contain large pixel parts as well as subpixel components. Some extraneous distortions other than translation, like rotation, zoom and out of focus distortions were unavoidable during recording. Thus, a twenty-frame section of the recorded video which contained minimal extraneous distortions are selected to form the Bilkent Sequence. The original frames of the sequence are necessarily interlaced, i.e., each frame consists of two fields that are displaced by 1 scan line vertically, and half a frame time temporally. Thus, for progressive processing of the Bilkent Sequence, the odd fields of each frame are discarded and the even fields are horizontally downsampled by a factor of 2. Six of these frames (frames with numbers 1, 5, 9, 13, 17, and 20) are shown in Figure 3.7 to show the content of the sequence and the amount of displacements among its frames.

The summation block size is chosen as  $100 \times 100$  and this block is located at the center-bottom of the Bilkent emblem.

Since the real displacements are not known for the Bilkent Sequence, the pixel and subpixel parts of the estimated displacements are plotted in this section to give an idea of the amount of misregistration in the sequence. The pixel part of the displacements are found using the phase correlation algorithm (Appendix A) prior to the application of the near-closed-form solution, phase and cross-correlation surface interpolation, and differentiation methods. Figure 3.8 shows the pixel part of the displacements which is found by the phase-correlation algorithm. Figure 3.9 compares the subpixel displacement estimates found using all five methods. We conclude from Figure 3.9 that the displacement estimates obtained using the near-closed-form solution is about the same for all the frames in the Bilkent Sequence as those obtained using the logarithmic search and the differentiation algorithm. When compared to the phase-correlation and cross-correlation surface interpolation algorithms, we observe that the displacement estimates of the near-closed-form solution are about the same on the average.

The performance of the near-closed-form solution in de-interlacing the original interlaced Bilkent Sequence is demonstrated in Figure 3.10. The top figure shows a sample interlaced frame, where the even and odd fields are relatively shifted due to the motion of the camcorder during recording. As can be seen from Figure 3.10, the image quality is very poor and the words are hardly legible. This problem can be resolved if the odd field is motion compensated with respect to the even field with subpixel accuracy. The bottom figure shows the same frame after motion compensation of the odd field with respect to the even field using the near-closed-form solution. A displacement of  $(d_1, d_2) = (5.6351, 0.4138)$  was found between the even and odd fields. As can be seen from Figure 3.10, there is a considerable improvement in the de-interlaced image. The words became legible and clearer.

Thus, interlaced sequences can also be processed without discarding any data and having to reduce the image size, as was done to generate the progressive Bilkent Sequence. In fact, we have applied the near-closed-form solution to the stabilization of the interlaced Bilkent Sequence as well. This is done as follows. First, the even fields of each frame are registered with subpixel

accuracy with respect to the even field of the reference frame. Then, the odd fields are registered with subpixel accuracy with respect to their corresponding even fields. We have observed that the application of the near closed-form solution resulted in steady image sequences in the cases of both the progressive and interlaced Bilkent Sequences. So, we can handle interlaced images without converting them to progressive and avoid the problem of reducing the image resolution.

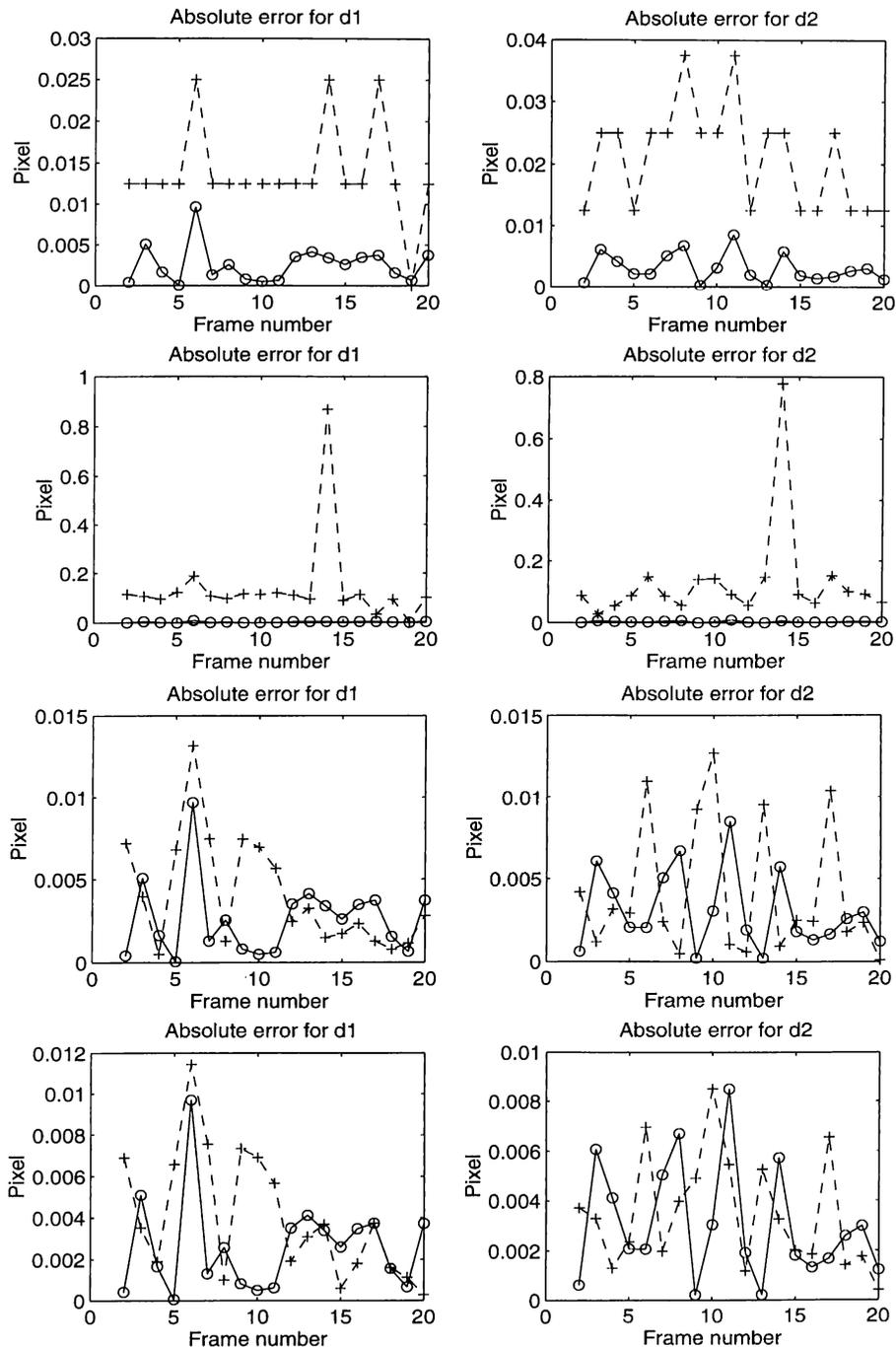


Figure 3.2: Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for Text Sequence with 20 dB SNR. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for  $d_1$ , while the second column compares the absolute errors for  $d_2$ .

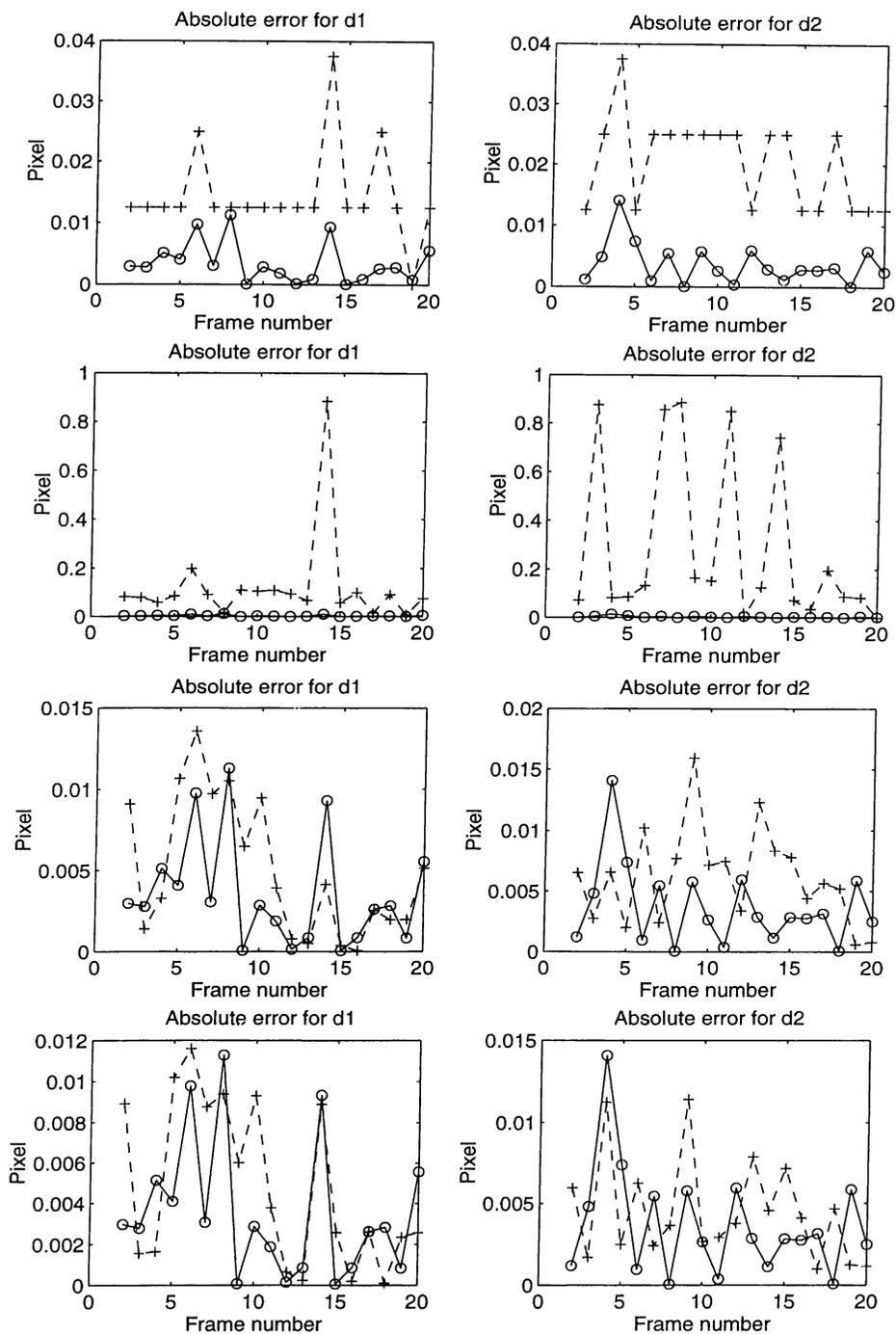


Figure 3.3: Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for Text Sequence with 10 dB SNR. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for  $d_1$ , while the second column compares the absolute errors for  $d_2$ .

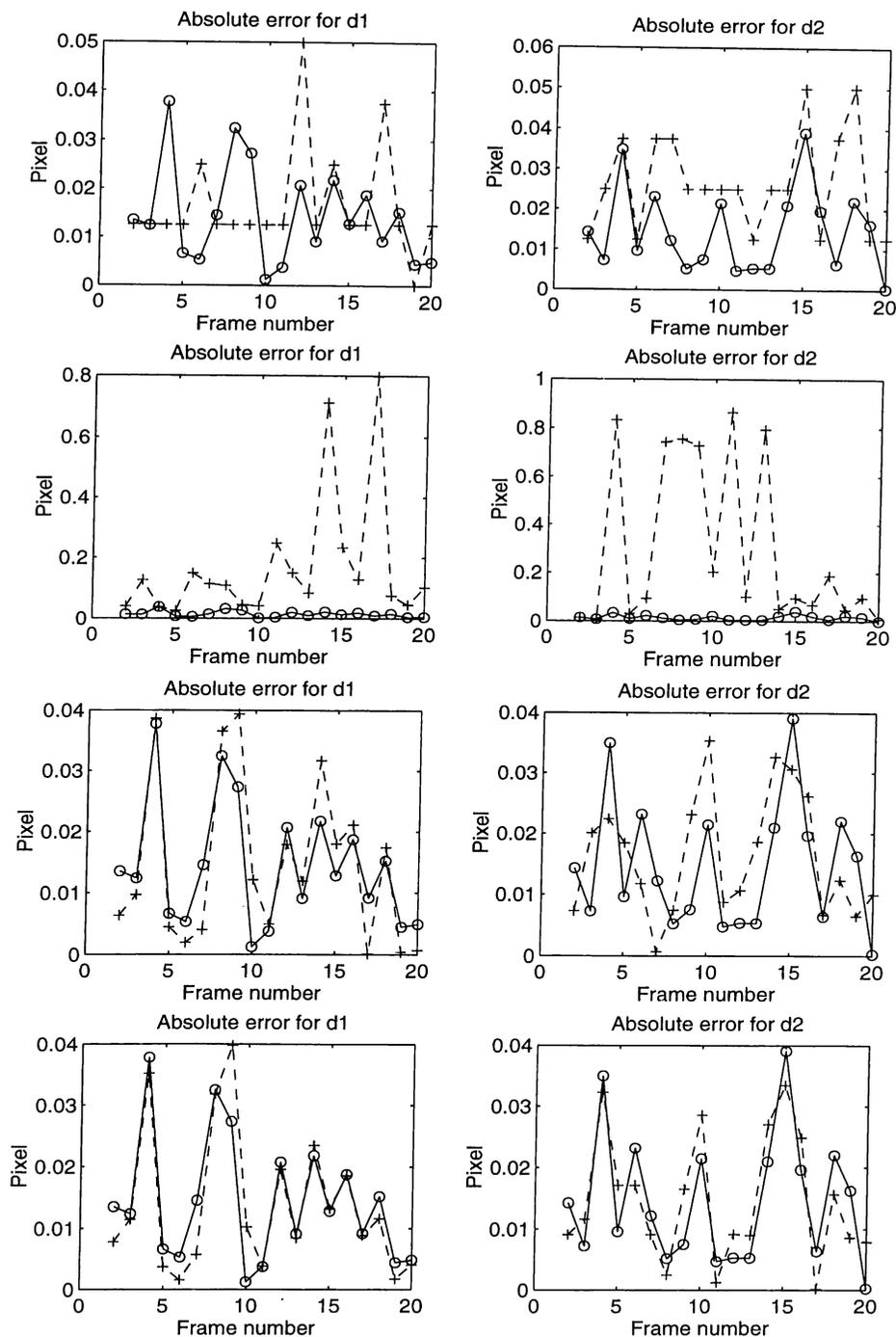


Figure 3.4: Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for Text Sequence with 5 dB SNR. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for  $d_1$ , while the second column compares the absolute errors for  $d_2$ .

	20 dB		10 dB		5 dB		CPU time
	$\epsilon(d_1)$	$\epsilon(d_2)$	$\epsilon(d_1)$	$\epsilon(d_2)$	$\epsilon(d_1)$	$\epsilon(d_2)$	
Near-closed-form	0.0026	0.0030	0.0035	0.0037	0.0143	0.0145	1.582
Logarithmic search	0.0138	0.0211	0.0145	0.0204	0.0164	0.0263	2.823
Phase correlation	0.1412	0.1293	0.1225	0.2897	0.1717	0.3012	1.504
Differentiation	0.0041	0.0042	0.0050	0.0062	0.0146	0.0163	1.653
Cross correlation	0.0040	0.0035	0.0048	0.0045	0.0137	0.0148	1.504

Table 3.1: The mean absolute displacement estimation errors  $\epsilon(d_1)$  and  $\epsilon(d_2)$  for Text Sequence. Block size is  $130 \times 60$ , CPU time (in seconds) includes the time spent for the detection of the pixel movement using the phase correlation algorithm.

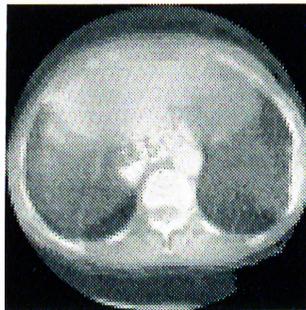


Figure 3.5: The first frame of the CT Sequence.

	CT		
	$\epsilon(d_1)$	$\epsilon(d_2)$	CPU time
Near-closed-form	0.0115	0.0099	0.678
Logarithmic search	0.0138	0.0230	1.268
Phase correlation	0.3184	0.3667	0.629
Differentiation	0.0128	0.0108	0.700
Cross correlation	0.3857	0.2366	0.629

Table 3.2: The mean absolute displacement estimation errors  $\epsilon(d_1)$  and  $\epsilon(d_2)$  over the 19 frames for the CT Sequence. Block size is  $60 \times 60$ , CPU time (in seconds) includes the time spent for the detection of the pixel movement using the phase correlation algorithm.

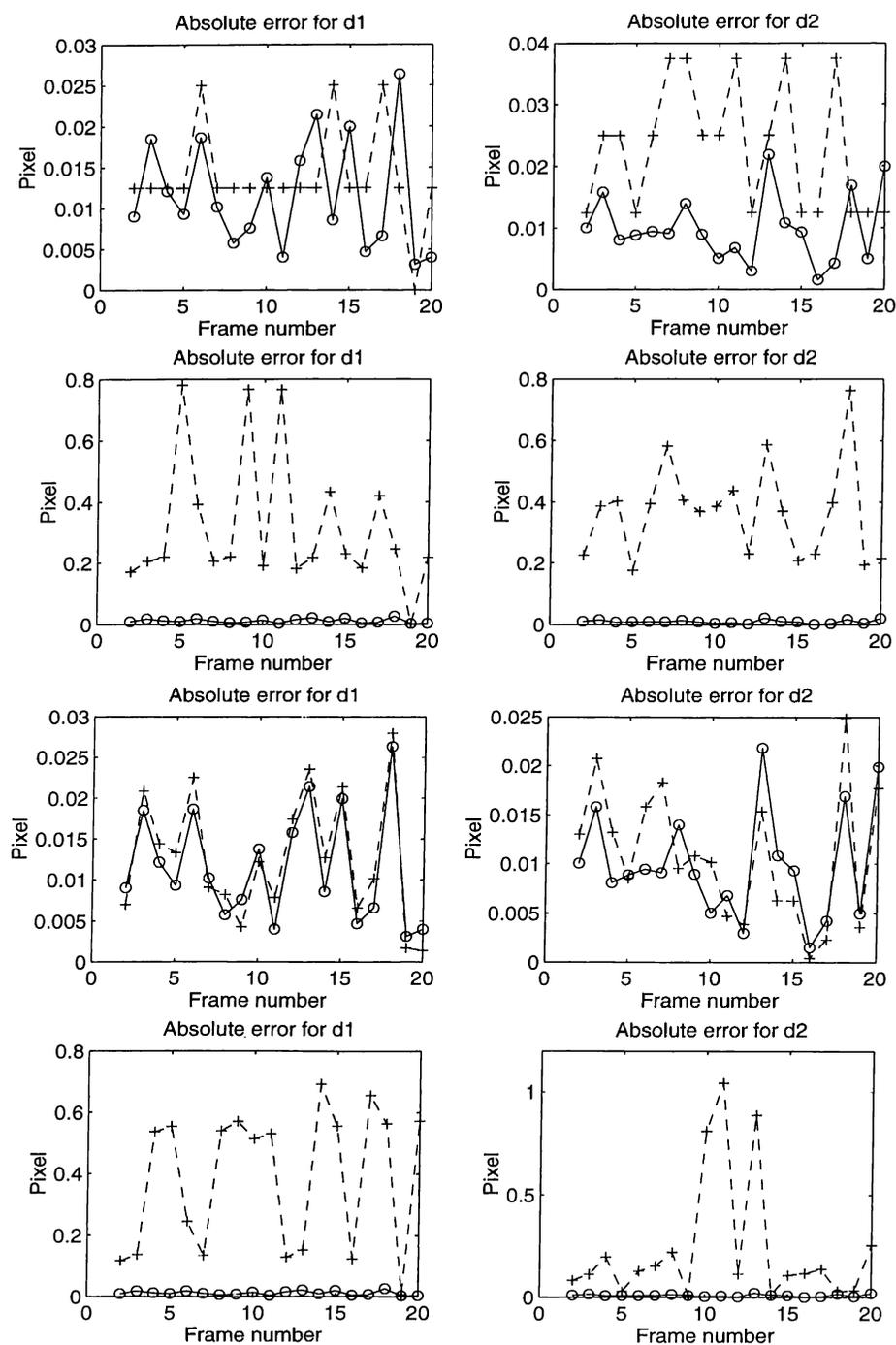


Figure 3.6: Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for CT Sequence. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for  $d_1$ , while the second column compares the absolute errors for  $d_2$ .



Figure 3.7: Six frames of the Bilkent Sequence: first and fifth frames (top row), ninth and thirteenth frames (second row), seventeenth and twentieth frames (third row).

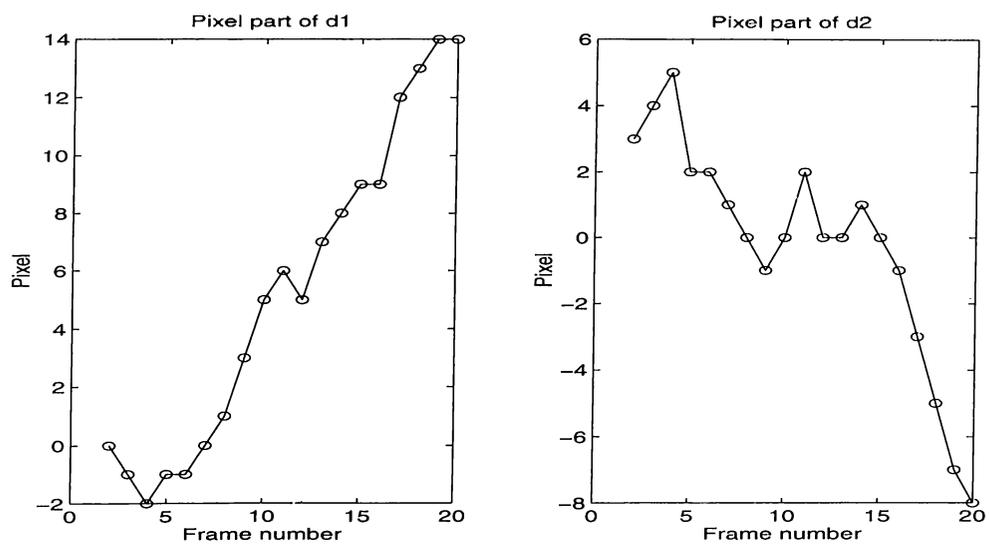


Figure 3.8: Pixel part of the displacements obtained by the phase correlation method for the “Bilkent” Sequence.

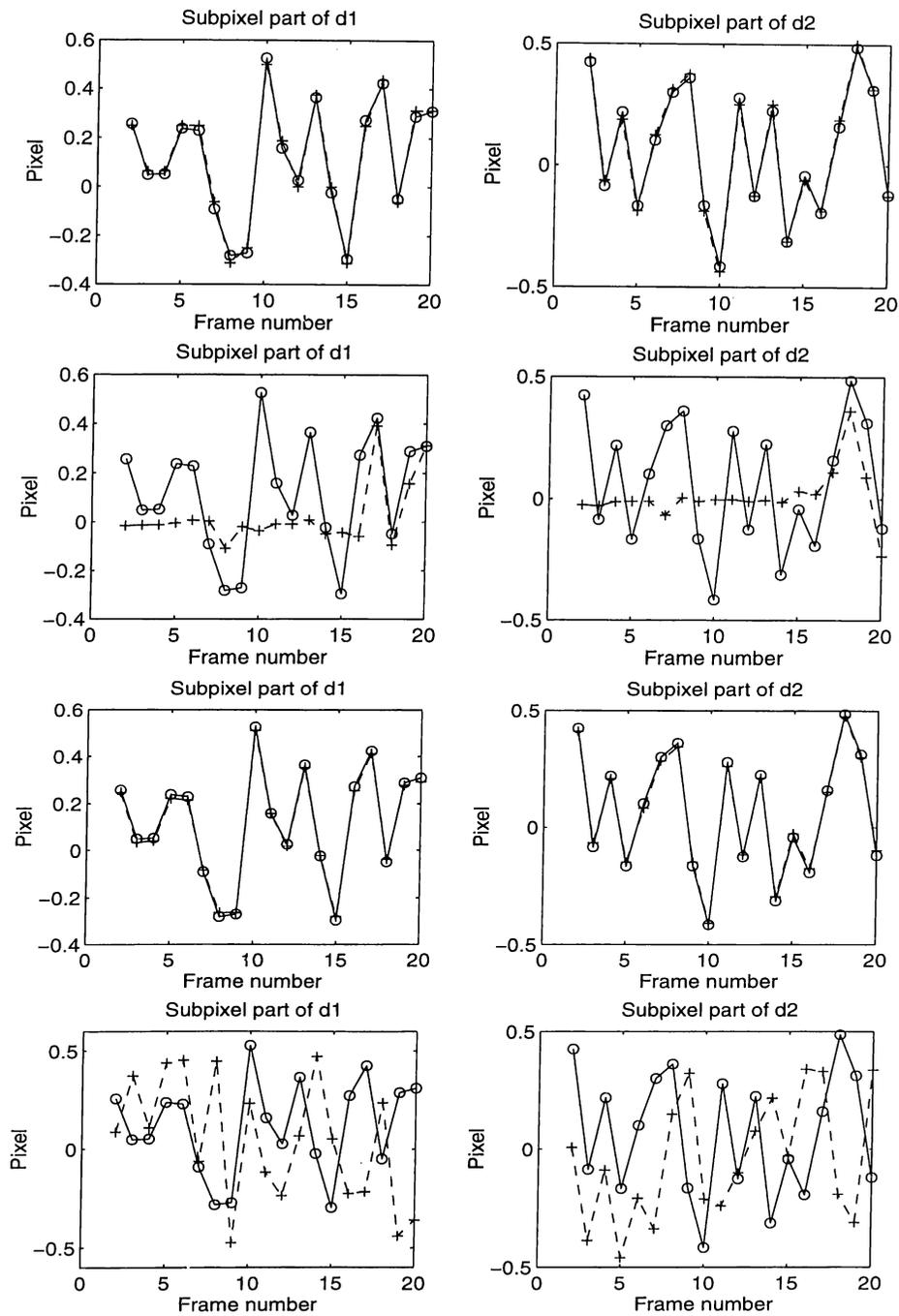


Figure 3.9: Plots of subpixel part of the displacements of the near-closed-form solution together with the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for “Bilkent” Sequence. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares  $d_1$ , while the second column compares  $d_2$ .

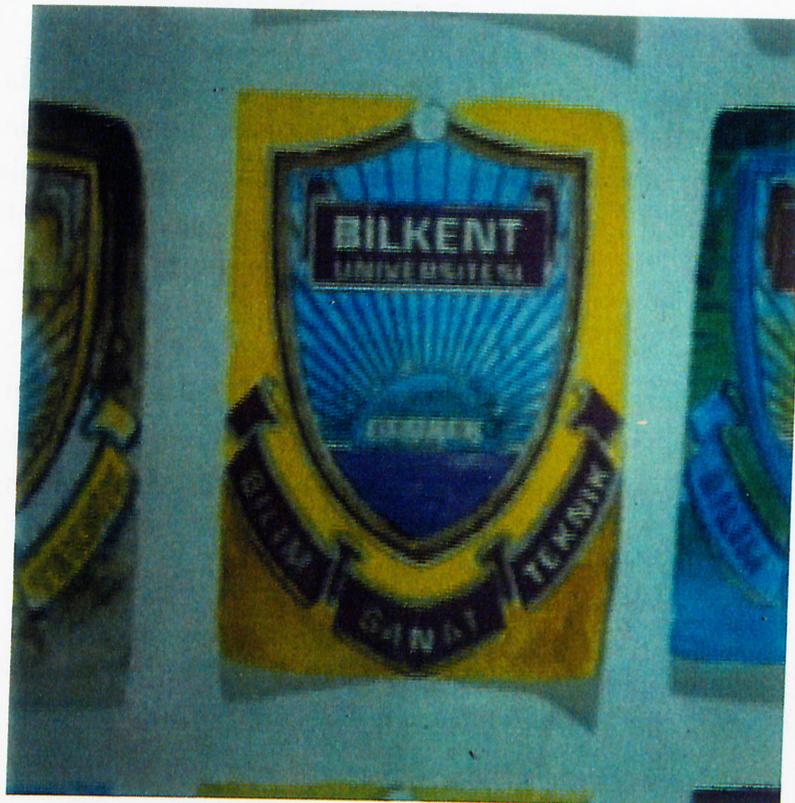


Figure 3.10: The figure at the top shows a frame of the Bilkent Sequence which is degraded due to the motion of the camera. The figure at the bottom shows the same frame after motion compensation of the odd field with respect the even field using the near closed-form solution.

## Chapter 4

# ACCOUNTING FOR INTENSITY VARIATIONS

### 4.1 Modeling

In estimating the subpixel displacement of a frame with respect to the reference frame, it is important to account for intensity variations, that are due to illumination changes, between the current frame and the reference frame. We assume that the intensity  $I_c$  of a pixel in the current frame is related to the intensity  $I_r$  in the reference frame by

$$I_c = \gamma I_r + \eta, \quad (4.1)$$

where  $\gamma$  and  $\eta$  are called the contrast and brightness parameters, respectively. Thus, in order to account for intensity variations, we modify the mean squared error expression given in (2.3) as

$$\text{MSE}^{(i)} = \frac{1}{N_1 N_2} \sum_{n_1, n_2 \in \mathcal{B}} [\gamma s_1(n_1, n_2) + \eta - \tilde{s}_2(n_1 + d_1, n_2 + d_2)]^2, \quad (d_1, d_2) \in \mathcal{Q}^{(i)}. \quad (4.2)$$

We note that in addition to  $d_1$  and  $d_2$ , two new parameters, namely  $\gamma$  and  $\eta$ , now need to be determined for each frame for the purpose of registration.

## 4.2 Method

We first find the optimal solution for  $\gamma$  and  $\eta$  in terms of  $d_1$  and  $d_2$  by setting  $\partial\text{MSE}^{(i)}/\partial\gamma = 0$  and  $\partial\text{MSE}^{(i)}/\partial\eta = 0$ . That is,

$$\begin{aligned}\frac{\partial\text{MSE}^{(i)}}{\partial\gamma} &= \sum_{n_1, n_2 \in \mathcal{B}} [\gamma s_1(n_1, n_2) + \eta - \tilde{s}_2(n_1 + d_1, n_2 + d_2)] s_1(n_1, n_2) = 0, \\ \frac{\partial\text{MSE}^{(i)}}{\partial\eta} &= \sum_{n_1, n_2 \in \mathcal{B}} [\gamma s_1(n_1, n_2) + \eta - \tilde{s}_2(n_1 + d_1, n_2 + d_2)] = 0.\end{aligned}\quad (4.3)$$

Note that these equations are linear in  $\gamma$  and  $\eta$ :

$$\begin{bmatrix} \sum s_1^2(n_1, n_2) & \sum s_1(n_1, n_2) \\ \sum s_1(n_1, n_2) & N_1 N_2 \end{bmatrix} \begin{bmatrix} \gamma \\ \eta \end{bmatrix} = \begin{bmatrix} \sum s_1(n_1, n_2) \tilde{s}_2(n_1 + d_1, n_2 + d_2) \\ \sum \tilde{s}_2(n_1 + d_1, n_2 + d_2) \end{bmatrix} \quad (4.4)$$

We repeat here the expression given in Chapter 2 for  $\tilde{s}_2(n_1 + d_1, n_2 + d_2)$  for convenience:

$$\tilde{s}_2(n_1 + d_1, n_2 + d_2) = S_0^{(i)} + S_1^{(i)} d_1 + S_2^{(i)} d_2 + S_3^{(i)} d_1 d_2, \quad (d_1, d_2) \in \mathcal{Q}^{(i)}, \quad (4.5)$$

where  $\mathcal{Q}^{(i)}$ ,  $i = 1, 2, 3, 4$ , denote the four quadrants. If we substitute the bilinear interpolation expression given in (4.5) for  $\tilde{s}_2(n_1 + d_1, n_2 + d_2)$  in (4.4), then the solution  $(\gamma^*, \eta^*)$  to 4.4 becomes a function of  $d_1$  and  $d_2$  in the following form

$$\begin{aligned}\gamma^* &= G_0^{(i)} + G_1^{(i)} d_1 + G_2^{(i)} d_2 + G_3^{(i)} d_1 d_2, \\ \eta^* &= H_0^{(i)} + H_1^{(i)} d_1 + H_2^{(i)} d_2 + H_3^{(i)} d_1 d_2,\end{aligned}\quad (4.6)$$

where the coefficients  $G_0^{(i)}, \dots, H_3^{(i)}$  are given in terms of the basic summations that are defined in Appendix C.1. The actual expressions for  $G_0^{(i)}, \dots, H_3^{(i)}$  in terms of the basic summations are provided in Appendix C.4. We note that the optimal values  $\gamma^*$  and  $\eta^*$  given in (4.6) are bilinear in  $d_1$  and  $d_2$ . Thus, when (4.5) and the expression for  $(\gamma^*, \eta^*)$  are substituted in (4.2), the expression within the square brackets will still be bilinear in  $d_1$  and  $d_2$ :

$$\begin{aligned}& G_0^{(i)} s_1(n_1, n_2) + H_0^{(i)} - S_0^{(i)} + [G_1^{(i)} s_1(n_1, n_2) + H_1^{(i)} - S_1^{(i)}] d_1 \\ & + [G_2^{(i)} s_1(n_1, n_2) + H_2^{(i)} - S_2^{(i)}] d_2 + [G_3^{(i)} s_1(n_1, n_2) + H_3^{(i)} - S_3^{(i)}] d_1 d_2.\end{aligned}\quad (4.7)$$

The above result is very important because when the intensity variations are incorporated, the new  $\text{MSE}^{(i)}$  still has the bilinear form as in (2.9):

$$\begin{aligned} \text{MSE}^{(i)} = & C_0^{(i)} + C_1^{(i)}d_1 + C_2^{(i)}d_2 + C_3^{(i)}d_1d_2 + C_4^{(i)}d_1^2 + C_5^{(i)}d_2^2 \\ & + C_6^{(i)}d_1^2d_2 + C_7^{(i)}d_1d_2^2 + C_8^{(i)}d_1^2d_2^2, \quad i = 1, 2, 3, 4. \end{aligned} \quad (4.8)$$

Of course, the expressions for the coefficients  $C_0^{(i)}, \dots, C_8^{(i)}$ , in terms of the basic summations will now be different than those used in Chapter 3. The new expressions for the coefficients  $C_0^{(i)}, \dots, C_8^{(i)}$ , are given in Appendix C.5. The expressions for the coefficients of the fifth degree polynomial

$$E_5^{(i)}d_2^5 + E_4^{(i)}d_2^4 + E_3^{(i)}d_2^3 + E_2^{(i)}d_2^2 + E_1^{(i)}d_2 + E_0^{(i)} = 0, \quad (4.9)$$

in terms of  $C_0^{(i)}, \dots, C_8^{(i)}$  will still be as given in Appendix C.3.

Thus, the incorporation of intensity variations into the near-closed-form solution is achieved by simply re-defining the coefficients  $C_0^{(i)}, \dots, C_8^{(i)}$ , in terms of the basic summations. The result is a novel near-closed-form solution that is insensitive to intensity variations. The difference of the new algorithm with that of Chapter 3 is only in the computation of the coefficients  $C_0^{(i)}, \dots, C_8^{(i)}$ ; otherwise the two algorithms are exactly the same.

If we model the intensity variations using only the brightness parameter, we will have the following MSE expression

$$\text{MSE}^{(i)} = \frac{1}{N_1N_2} \sum_{n_1, n_2 \in \mathcal{B}} [s_1(n_1, n_2) + \eta - \tilde{s}_2(n_1 + d_1, n_2 + d_2)]^2, \quad (d_1, d_2) \in \mathcal{Q}^{(i)}. \quad (4.10)$$

Then, using procedure given above, the optimal brightness parameter  $\eta^*$  is found to be

$$\eta^* = \frac{1}{N_1N_2} \sum \tilde{s}_2(n_1 + d_1, n_2 + d_2) - \frac{1}{N_1N_2} \sum s_1(n_1, n_2). \quad (4.11)$$

Since the contrast parameter  $\gamma$  is assumed to be 1, we will have

$$G_0^{(i)} = 1, \quad \text{and} \quad G_1^{(i)} = G_2^{(i)} = G_3^{(i)} = 0. \quad (4.12)$$

The coefficients  $H_1^{(i)}, H_2^{(i)}, H_3^{(i)}, H_4^{(i)}$  for the brightness parameters are given in Appendix C.6. The coefficients of the MSE expression (4.8) for this case can be obtained by substituting  $H_1^{(i)}, H_2^{(i)}, H_3^{(i)}, H_4^{(i)}$  and (4.12) in Appendix C.5.

If we model the intensity variations using only the contrast parameter, we will have the following MSE expression

$$\text{MSE}^{(i)} = \frac{1}{N_1 N_2} \sum_{n_1, n_2 \in \mathcal{B}} [\gamma s_1(n_1, n_2) - \tilde{s}_2(n_1 + d_1, n_2 + d_2)]^2, \quad (d_1, d_2) \in \mathcal{Q}^{(i)}. \quad (4.13)$$

Then, using procedure given above, the optimal contrast parameter  $\gamma^*$  is found to be

$$\gamma^* = \frac{\sum \tilde{s}_2(n_1 + d_1, n_2 + d_2) s_1(n_1, n_2)}{\sum s_1^2(n_1, n_2)}. \quad (4.14)$$

Since the contrast parameter *eta* is assumed to be 0, we will have

$$H_0^{(i)} = H_1^{(i)} = H_2^{(i)} = H_3^{(i)} = 0. \quad (4.15)$$

The coefficients  $G_1^{(i)}, G_2^{(i)}, G_3^{(i)}, G_4^{(i)}$  for the contrast parameters are given in Appendix C.7. The coefficients of the MSE expression (4.8) for this case can be obtained by substituting  $G_1^{(i)}, G_2^{(i)}, G_3^{(i)}, G_4^{(i)}$  and (4.15) in Appendix C.5.

Thus the steps of the proposed algorithm in case of intensity variations can be summarized as follows:

1. Compute the basic summations  $A_{0,0}, A_{0,0;0,0}, B_{0,0; i,j}, D_{i,j}, D_{i,j; k,\ell}$ , given in Appendix C.2 over a specified block of pixels.
2. Compute the coefficients for contrast and brightness parameters,  $C_0^{(i)}, \dots, H_3^{(i)}$ , given in Appendix C.4 (or Appendix C.6 or Appendix C.7 depending on the intensity variation model) for each quadrant.
3. Compute the MSE coefficients  $C_0^{(i)}, \dots, C_8^{(i)}$ , given in Appendix C.5 for each quadrant, i.e., for each  $i = 1, 2, 3, 4$ .
4. Compute the coefficients  $E_0, \dots, E_5$ , of the fifth order polynomial in Appendix C.3 for each quadrant.
5. Find the zeroes of (4.9) for each quadrant. Among the acceptable ones, pick the one which gives the minimum MSE. That gives the near-closed-form solution. If there is no acceptable solution, find  $(d_1, d_2)$  that minimizes the MSE expression (4.8), using the efficient search method of Chapter 2 using the new definition for the coefficients  $C_0^{(i)}, \dots, C_8^{(i)}$  proposed in this chapter.

### 4.3 Results

The simulations are done using five synthetically generated sequences and a real sequence. The synthetic sequences are called Text-1, Text-2, Text-3, Text-4, and CT-2 Sequences. The real sequence is called the Bilkent-2 Sequence, which again contains the Bilkent emblem. This sequence is recorded using a camcorder as described in Chapter 3. Each sequence consists of 20 frames. The Text and CT Sequences contain simulated intensity variations while the Bilkent-2 Sequence contains natural intensity variations that are created by varying the brightness of the light source during recording.

The Text Sequences are generated from the “Text” image as described in Chapter 2, where the PSNR is chosen as 10 dB. In order to introduce the intensity variations to the subsequent frames of a synthetically generated sequence, the pixel intensities in the frames are first multiplied by  $\gamma$ , and  $\eta$  is added afterwards as explained in (4.1). If we let  $k$  denote the frame number, the values of  $\gamma$  and  $\eta$  chosen for the  $k^{\text{th}}$  frame are defined in our experiments as

$$\gamma(k) = 1 + \gamma_0 k \quad (4.16)$$

$$\eta(k) = \eta_0 k \quad (4.17)$$

in terms of the contrast and brightness variation factors  $\gamma_0$  and  $\eta_0$ .

The intensity variations in the Text Sequences are simulated by varying both  $\gamma$  and  $\eta$  parameters (Text-2 Sequence), varying only the  $\gamma$  parameter (Text-3 Sequence), and varying only the  $\eta$  parameter (Text-4 Sequence) using the rule given in (4.17). The reference and last frames of the Text Sequences used in the simulations are shown in Figure 4.1. Note that the Text-1 Sequence does not contain any intensity variations.

The other synthetically generated sequence used in the simulations is the CT-2 Sequence, which contains both contrast and brightness variations. The first frame of the CT-2 Sequence is the same as that of the CT Sequence introduced in Chapter 3.2.2. The last frame of the CT-2 Sequence is shown in Figure 4.2.

The proposed algorithm is also tested using a real colored sequence, namely, the Bilkent-2 Sequence. Six frames of the sequence are shown in Figure 4.3 to show the degree of intensity variations and the amount of displacements among the frames. They are the first and seventh frames (top row), thirteenth and fifteenth frames (second row), and seventeenth and twentieth frames (third row).

In the following, the performance of the proposed near-closed-form solution with compensation to intensity variations between frames is compared with the popular subpixel displacement estimation methods existing in the literature. The methods that the near-closed-form solution is compared against are, the logarithmic search method (Appendix B.2), the differentiation method (Appendix B.3), and the phase-correlation and cross-correlation surface interpolation methods (Appendix B.1). We note that a  $7 \times 7$  uniform blur is applied to all images in the Text Sequences prior to displacement estimation in order to reduce the effects of bilinear interpolation and the additive noise. The extent of the blur is chosen as  $5 \times 5$  for the CT and Bilkent Sequences since they contains less noise than the Test Sequences. The accuracy of the logarithmic search is chosen as  $1/16$ , i.e., 0.0625, pixels.

In all of the following experiments, the near-closed-form solution that takes into account both the contrast and brightness variations is employed.

### 4.3.1 Text-1 Sequence

Text-1 Sequence contains no intensity variations. Nevertheless, the near-closed-form solution which makes illumination compensation for both  $\gamma$  and  $\eta$  is applied here. Figure 4.4 gives a comparison of absolute displacement estimation errors of the near-closed-form solution with other methods. Figure 4.5 gives the plot of  $\gamma$  and  $\eta$  values found by the near-closed-form solution. As can be seen from Figure 4.4, the absolute displacement estimation error for the near-closed-form solution is smaller than that of logarithmic search and phase correlation interpolation methods for almost all the frames. Note that these results are similar to the ones given in Figure 3.3. As seen from Figure 4.5, the estimated  $\gamma$  values, i.e., the multiplicative intensity variation component

is very close to 1 as expected. Similarly, the estimated  $\eta$  values are around 0 which is again the expected result. The variations in the estimated  $\eta$  values may be caused by the additive white Gaussian noise present in the images.

### 4.3.2 Text-2 Sequence

Text-2 Sequence contains both additive and multiplicative intensity variations. The pair  $(\gamma_0, \eta_0)$  is chosen as  $(-0.02, -2)$ . When this sequence is registered with the algorithm proposed in Chapter 3, which does not employ any compensation for the intensity variations, it is observed that the absolute error of the closed form solution starts to increase with increasing frame number, i.e., with increasing intensity variation with respect to the reference frame. The algorithm of Chapter 3 falls back to the efficient search method for the last few frames of the Text-2 Sequence because it can not find a closed-form solution.

Figure 4.6 gives a comparison of absolute displacement estimation errors of the near-closed-form solution with other methods when illumination compensation for both  $\gamma$  and  $\eta$  is employed. It can be seen that absolute displacement estimation error for the near-closed-form solution is much smaller than that of the logarithmic search, phase-correlation interpolation and differentiation methods for almost all the frames. Note that in the differentiation method, the absolute error for  $d_1$  increases with increasing frame number, which means that the method is not robust to intensity variations between the frames. The performance of the cross-correlation interpolation method is close to the performance of the near-closed-form solution. Figure 4.7 gives the plot of  $\gamma$  and  $\eta$  values found by the near-closed-form solution. It can be seen that the near-closed-form solution is able to detect the parameters  $\gamma$  and  $\eta$  almost perfectly. Only at the twentieth frame, a small deviation from actual values have occurred.

### 4.3.3 Text-3 Sequence

Text-3 Sequence contains only multiplicative intensity variations, i.e., the pair  $(\gamma_0, \eta_0)$  is chosen as  $(-0.04, 0)$ . Figure 4.8 gives a comparison of absolute displacement estimation errors of the near-closed-form solution with other methods when illumination compensation for both  $\gamma$  and  $\eta$  is employed. It can be seen from the figure that the absolute displacement estimation error is much smaller than that of the logarithmic search, phase correlation interpolation and differentiation methods for almost all the frames. The differentiation method starts to give large errors with increasing frame number. The performance of the cross correlation interpolation method is close to the performance of the near-closed-form solution. It is observed that the near-closed-form solution also gives good results when only compensation for  $\gamma$  is allowed. However, it is also observed that if illumination compensation is allowed for both  $\gamma$  and  $\eta$ , one may obtain better results even if only a contrast variation is actually present on the images. This probably results from the additive noise existing on the frames. The use of the additive parameter  $\eta$  may actually compensate for the effects of additive noise on the results. Figure 4.9 gives the plot of  $\gamma$  and  $\eta$  values found by the near-closed-form solution. The  $\gamma$  value for each frame is almost perfectly found for each frame. The estimated  $\eta$  parameter takes values around 0 as expected.

### 4.3.4 Text-4 Sequence

Text-4 Sequence contains only additive intensity variations, i.e., the pair  $(\gamma_0, \eta_0)$  is chosen as  $(0, -3)$ . Figure 4.10 gives a comparison of absolute displacement estimation errors of the near-closed-form solution with other methods when illumination compensation for both  $\gamma$  and  $\eta$  is employed. It can be seen from Figure 4.10 that the absolute displacement estimation error is much smaller than that of the logarithmic search, phase-correlation interpolation and differentiation methods for almost all the frames. The differentiation method starts to give large errors with increasing frame number for  $d_1$ . The performance of the cross-correlation interpolation method is close to the performance of the near-closed-form solution. Figure 4.11 gives the plot of  $\gamma$  and  $\eta$  values found by

the near-closed-form solution. The  $\eta$  value for each frames is almost perfectly found for each frame. The estimated  $\gamma$  parameter takes values around 1 as expected.

### 4.3.5 CT-2 Sequence

The CT-2 Sequence contains both additive and multiplicative intensity variations, i.e., the pair  $(\gamma_0, \eta_0)$  is chosen as  $(-0.02, -2)$ . Figure 4.12 gives a comparison of absolute displacement estimation errors of the near-closed-form solution with other methods. It is seen that the absolute displacement estimation error of the near-closed-form solution is much smaller than the other methods. Note, especially, the abrupt increase in the error of the differentiation method for  $d_1$ . Also, the performance of the cross-correlation method is not as good as it is in the Text Sequences. Figure 4.13 gives the plot of  $\gamma$  and  $\eta$  values found by the near-closed-form solution. The near-closed-form solution tracks the actual  $\gamma$  and  $\eta$  parameters quite closely for each frame.

### 4.3.6 Bilkent-2 Sequence

Bilkent-2 Sequence is a real colored sequence with intensity variations of unknown type. The pixel parts of the estimated displacements values are given in Figure 4.14 to give an idea about the amount of misregistration among the frames. The pixel part of the displacement with respect to the reference frame takes a maximum value of 16 pixels in the horizontal direction, i.e., for  $d_1$ . Figure 4.15 gives plots of the subpixel part of the estimated displacements obtained with the near-closed-form solution as well as the other methods. Although there exists some differences between the estimated displacement values, the values are about the same on the average for all five methods. In Figure 4.16, the estimated values for the contrast and brightness parameters  $\gamma$  and  $\eta$  are plotted. It can be seen that, both additive and multiplicative illumination variations exist between the frames of the Bilkent-2 Sequence. The  $\gamma$  takes a minimum of approximately 0.37 and  $\eta$  can take values as small as approximately -22. These correspond to the darkest frames in the sequence.

### 4.3.7 Summary of Results

In order to see the results compactly, Tables 4.1 and 4.2 are provided which show the mean of absolute errors over 19 frames for the Text Sequences with 10 dB noise level. It can be seen that the mean of mean absolute errors for the near-closed-form solution is smaller than the other algorithms except the cross-correlation interpolation method. The performances of the near-closed-form solution and the cross-correlation interpolation methods are close to each other for the Text Sequences. A comparison of the CPU times is also provided in Table 4.2. The slowest method turns out to be the logarithmic search and all other methods are closed to each other in terms of their CPU times.

	Text-1		Text-2	
	$\epsilon(d_1)$	$\epsilon(d_2)$	$\epsilon(d_1)$	$\epsilon(d_2)$
Near-closed-form	0.0040	0.0051	0.0032	0.0053
Logarithmic search	0.0145	0.0211	0.1717	0.2467
Phase correlation	0.0823	0.2024	0.0891	0.0739
Differentiation	0.0059	0.0045	0.0217	0.0083
Cross correlation	0.0058	0.0033	0.0056	0.0047

Table 4.1: The mean absolute displacement estimation errors  $\epsilon(d_1)$  and  $\epsilon(d_2)$ .

	Text-3		Text-4		CPU time
	$\epsilon(d_1)$	$\epsilon(d_2)$	$\epsilon(d_1)$	$\epsilon(d_2)$	
Near-closed-form	0.0081	0.0091	0.0044	0.0051	1.559
Logarithmic search	0.1914	0.2664	0.0138	0.0230	2.692
Phase correlation	0.1754	0.1991	0.1375	0.1250	1.476
Differentiation	0.0371	0.0313	0.0167	0.0039	1.620
Cross correlation	0.0095	0.0085	0.0034	0.0038	1.476

Table 4.2: The mean absolute displacement estimation errors  $\epsilon(d_1)$  and  $\epsilon(d_2)$ .

A summary of results is also provided for the CT-2 Sequence in Table 4.3. The superiority of the near-closed-form solution over the other methods is seen clearly in this table. The near-closed-form solution is far better than the cross-correlation interpolation method for this case.

	CT	
	$\epsilon(d_1)$	$\epsilon(d_2)$
Near-closed-form	0.0084	0.0075
Logarithmic search	0.4079	0.1408
Phase correlation	0.3566	0.3748
Differentiation	0.5720	0.0359
Cross correlation	0.5714	0.1982

Table 4.3: The mean absolute displacement estimation errors  $\epsilon(d_1)$  and  $\epsilon(d_2)$  for the CT-2 Sequence containing intensity variations.

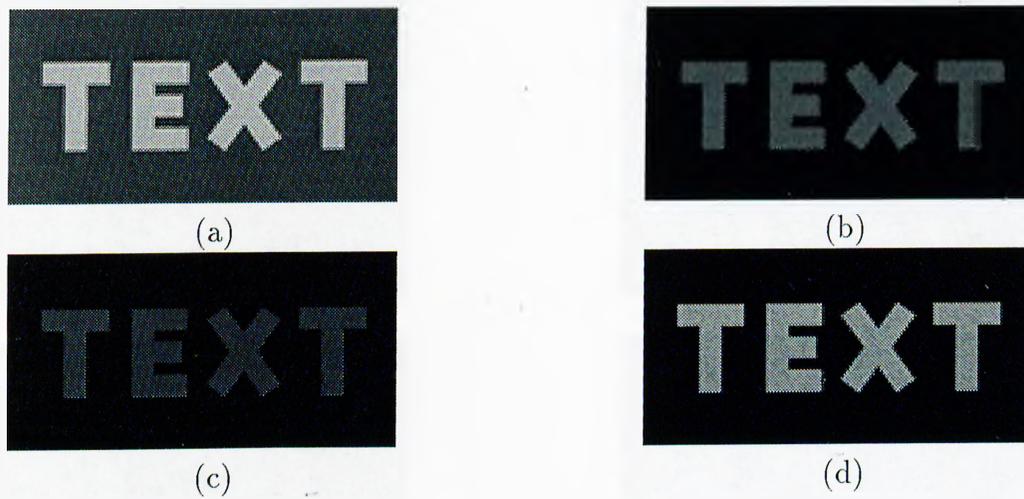


Figure 4.1: (a) The reference frame of all Text Sequences. (b) The last frame of the Text-2 Sequence. (c) The last frame of the Text-3 Sequence. (d) The last frame of the Text-4 Sequence

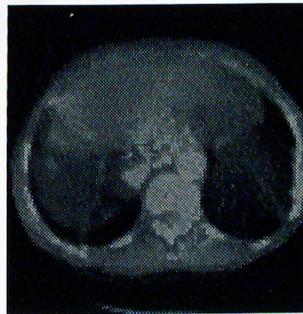


Figure 4.2: The last frame of the CT-2 Sequence generated with  $(\gamma_0, \eta_0) = (-0.02, -2)$



Figure 4.3: Six frames of the Bilkent-2 Sequence which contains intensity variations. First and seventh frames (top row), thirteenth and fifteenth frames (second row), seventeenth and twentieth frames (third row).

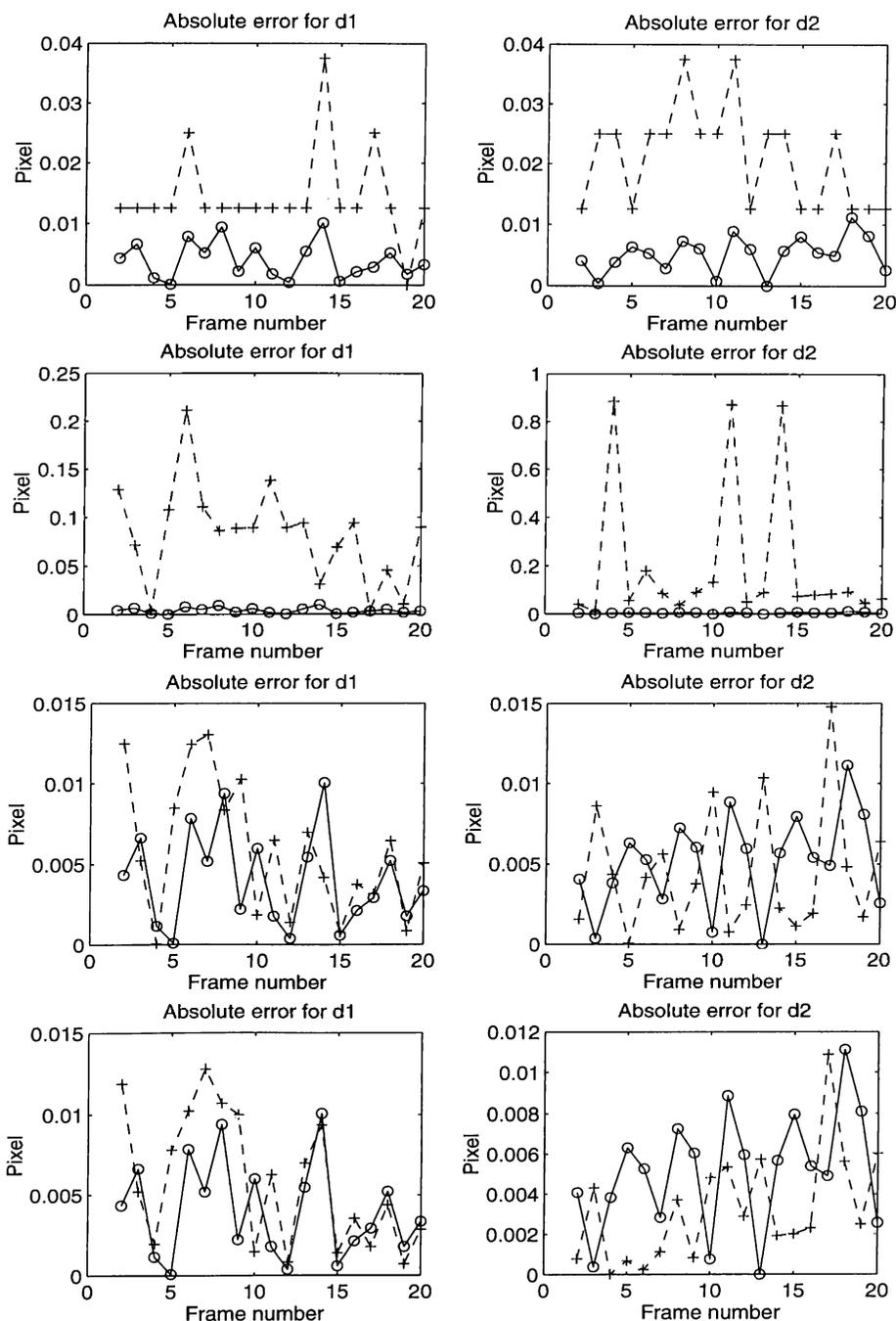


Figure 4.4:  $(\gamma_0, \eta_0) = (0, 0)$ . Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for Text-1 Sequence with 10dB SNR. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for  $d_1$ , while the second column compares the absolute errors for  $d_2$ .

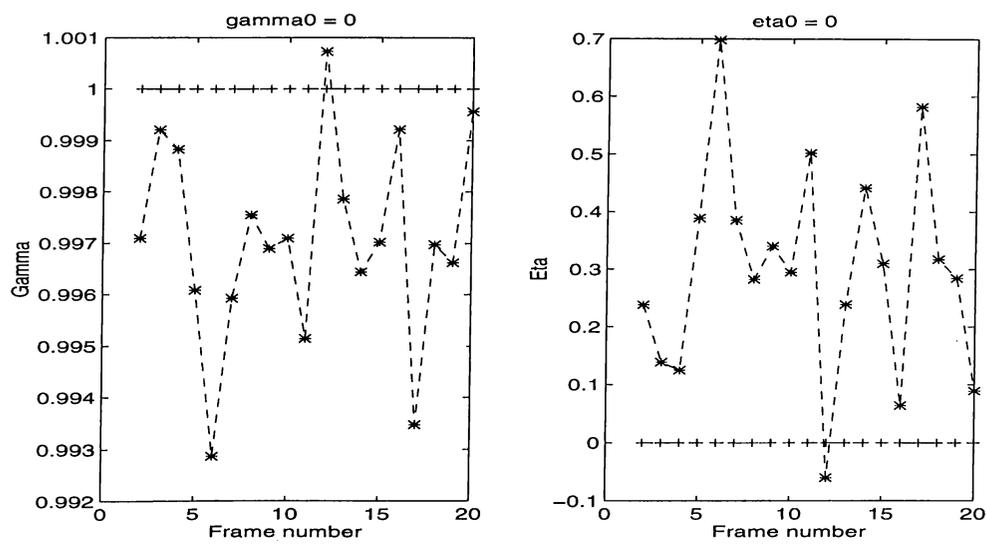


Figure 4.5: The  $\gamma$  and  $\eta$  values found for the Text-1 Sequence which does not contain any intensity variations. The sign “star” and “+” signs denote the estimated and true values for  $\gamma$  (on the left) and  $\eta$  (on the right) respectively.

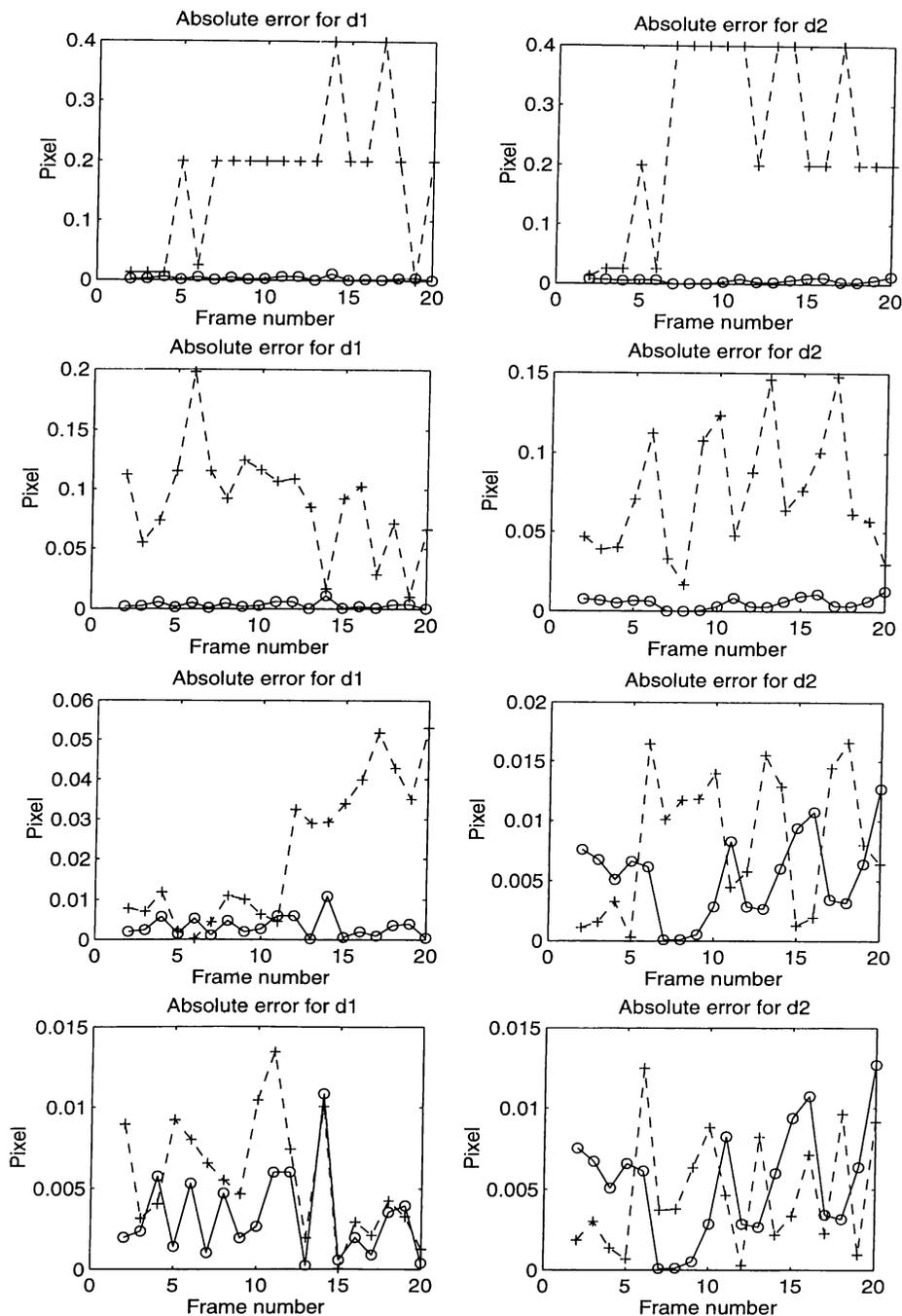


Figure 4.6:  $(\gamma_0, \eta_0) = (-0.02, -2)$ . Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for Text-2 Sequence with 10dB SNR. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for  $d_1$ , while the second column compares the absolute errors for  $d_2$ .

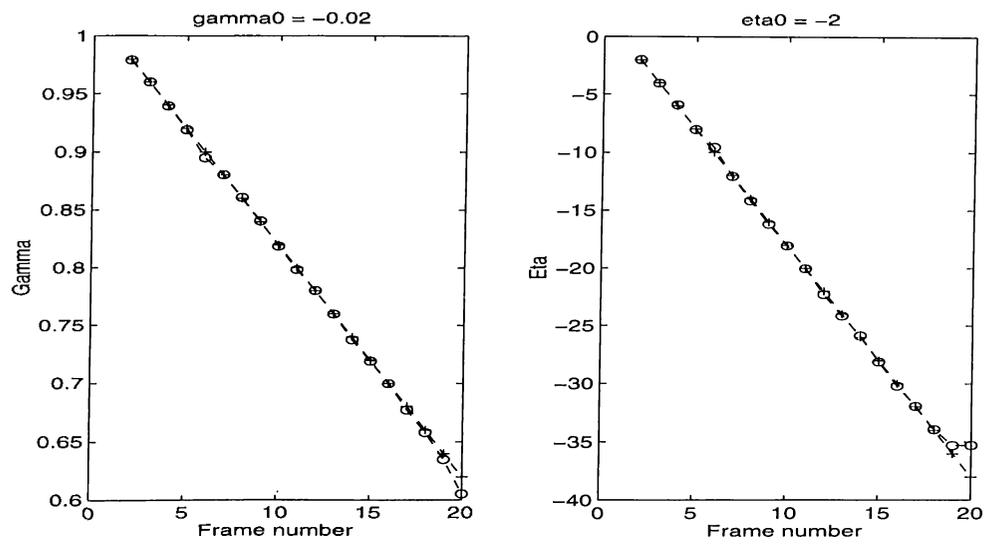


Figure 4.7: The  $\gamma$  and  $\eta$  values found for the Text-2 Sequence in which  $(\gamma_0, \eta_0) = (-0.02, -2)$ . The “\*” and “+” signs denote the estimated and true values for  $\gamma$  (on the left) and  $\eta$  (on the right) respectively.

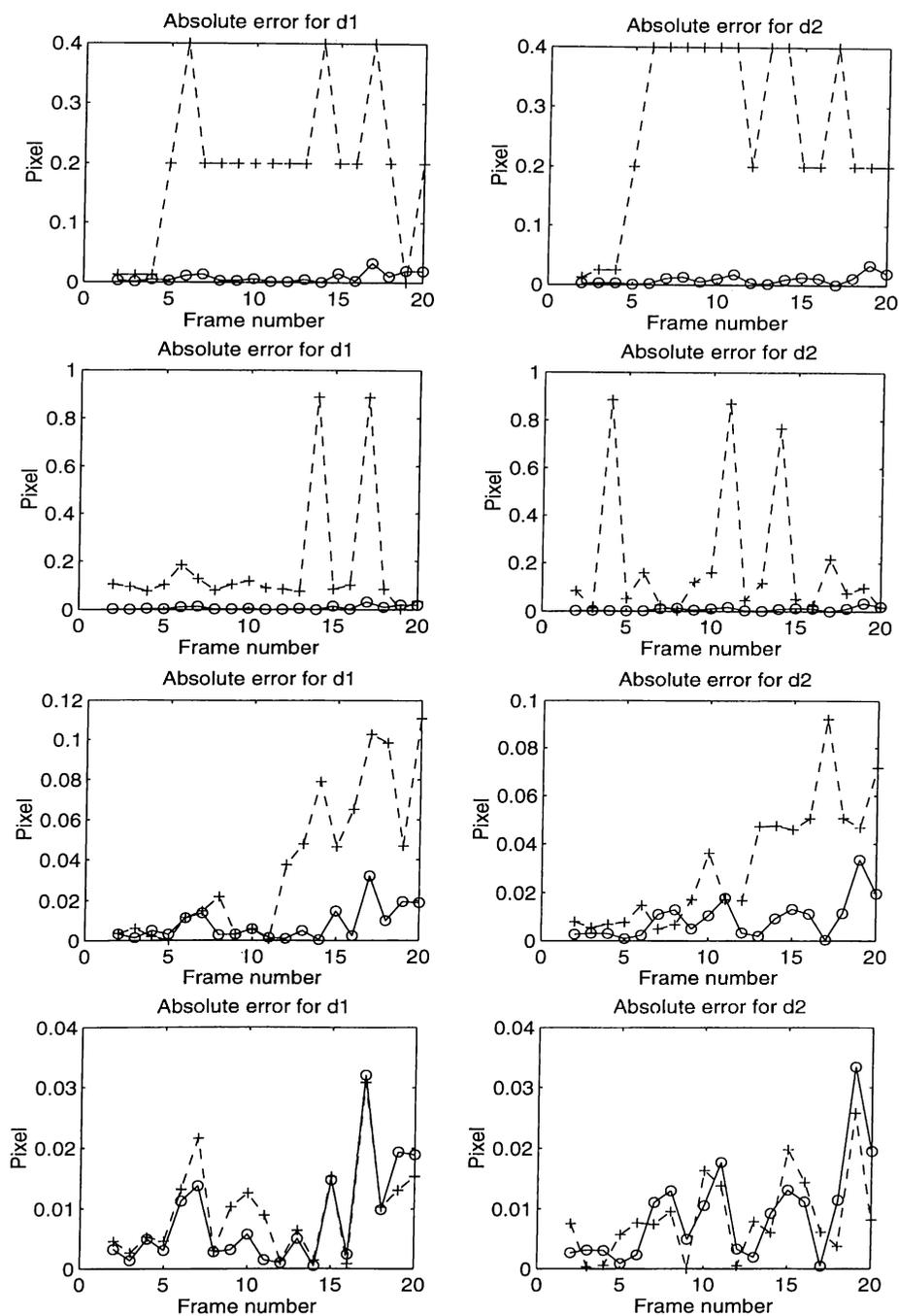


Figure 4.8:  $(\gamma_0, \eta_0) = (-0.04, 0)$ . Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for 'Text-3 Sequence with 10dB SNR. "Circles" denote the results obtained by the near-closed-form solution, while "+" signs denote the results obtained by the other methods. The first column compares the absolute errors for  $d_1$ , while the second column compares the absolute errors for  $d_2$ .

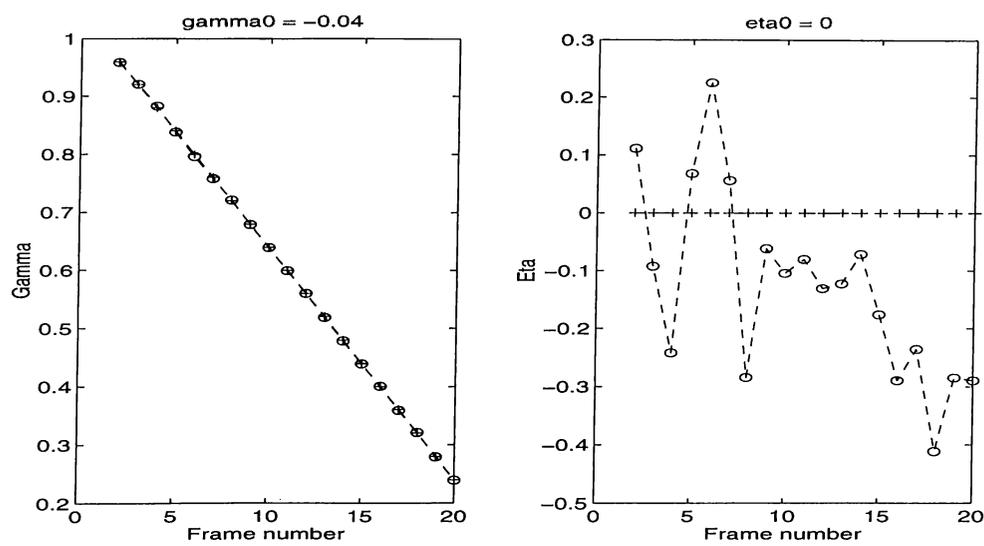


Figure 4.9: The  $\gamma$  and  $\eta$  values found for the Text-3 Sequence in which  $(\gamma_0, \eta_0) = (-0.04, 0)$ . The “\*” and “+” signs denote the estimated and true values for  $\gamma$  (on the left) and  $\eta$  (on the right) respectively.

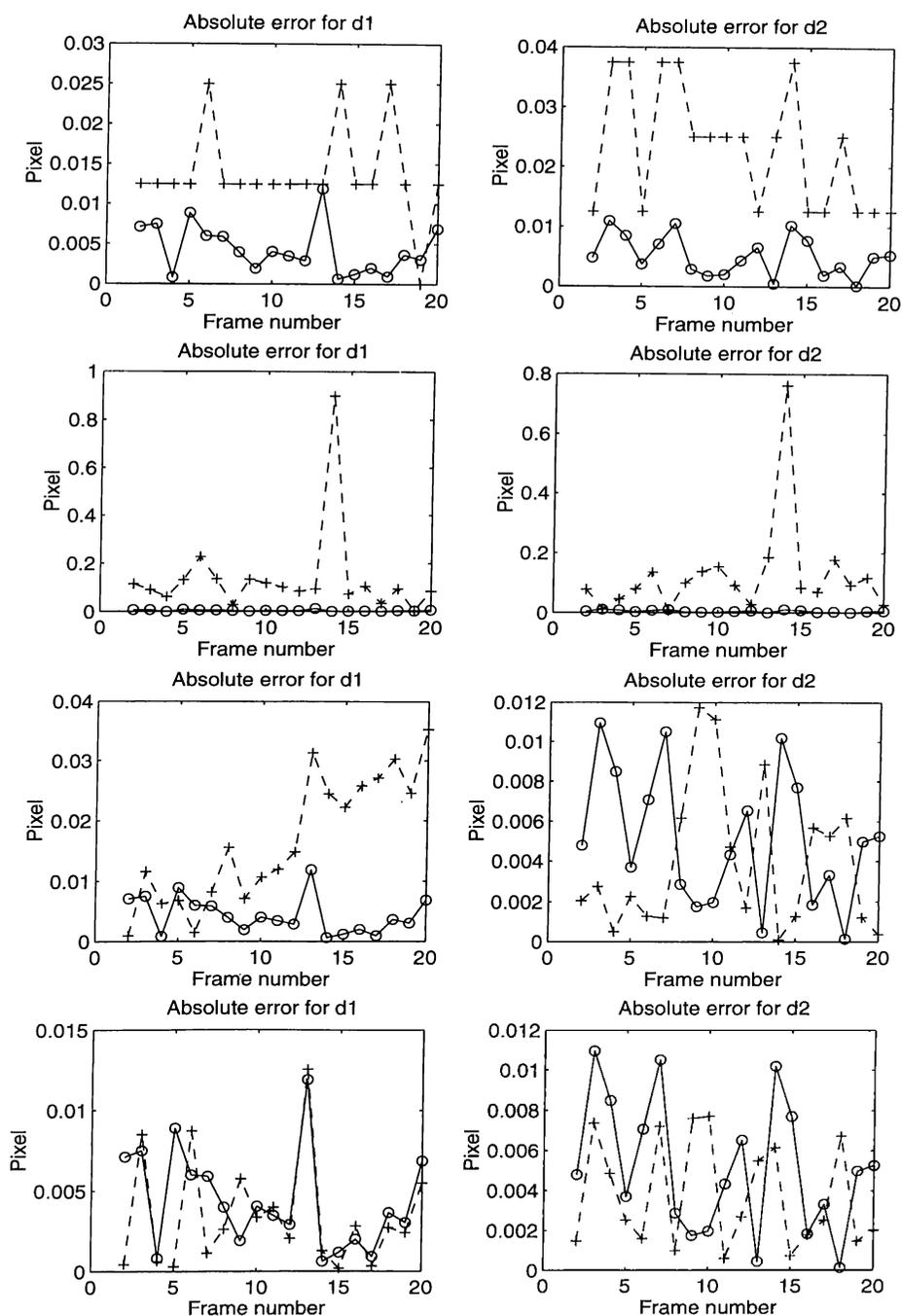


Figure 4.10:  $(\gamma_0, \eta_0) = (0, -3)$ . Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for 'Text-4 Sequence with 10dB SNR. "Circles" denote the results obtained by the near-closed-form solution, while "+" signs denote the results obtained by the other methods. The first column compares the absolute errors for  $d_1$ , while the second column compares the absolute errors for  $d_2$ .

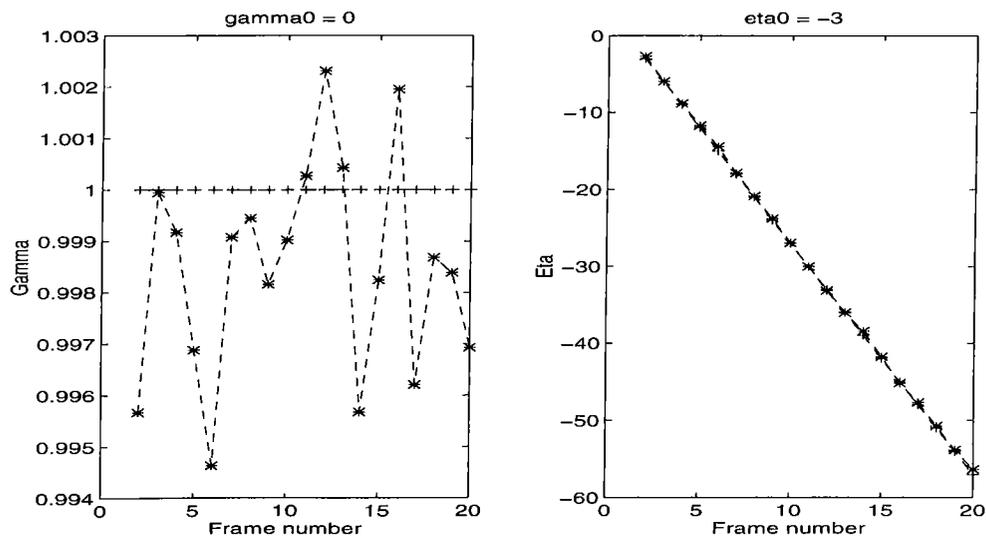


Figure 4.11: The  $\gamma$  and  $\eta$  values found for the Text-3 Sequence in which  $(\gamma_0, \eta_0) = (0, -3)$ . The “\*” and “+” signs denote the estimated and true values for  $\gamma$  (on the left) and  $\eta$  (on the right) respectively.

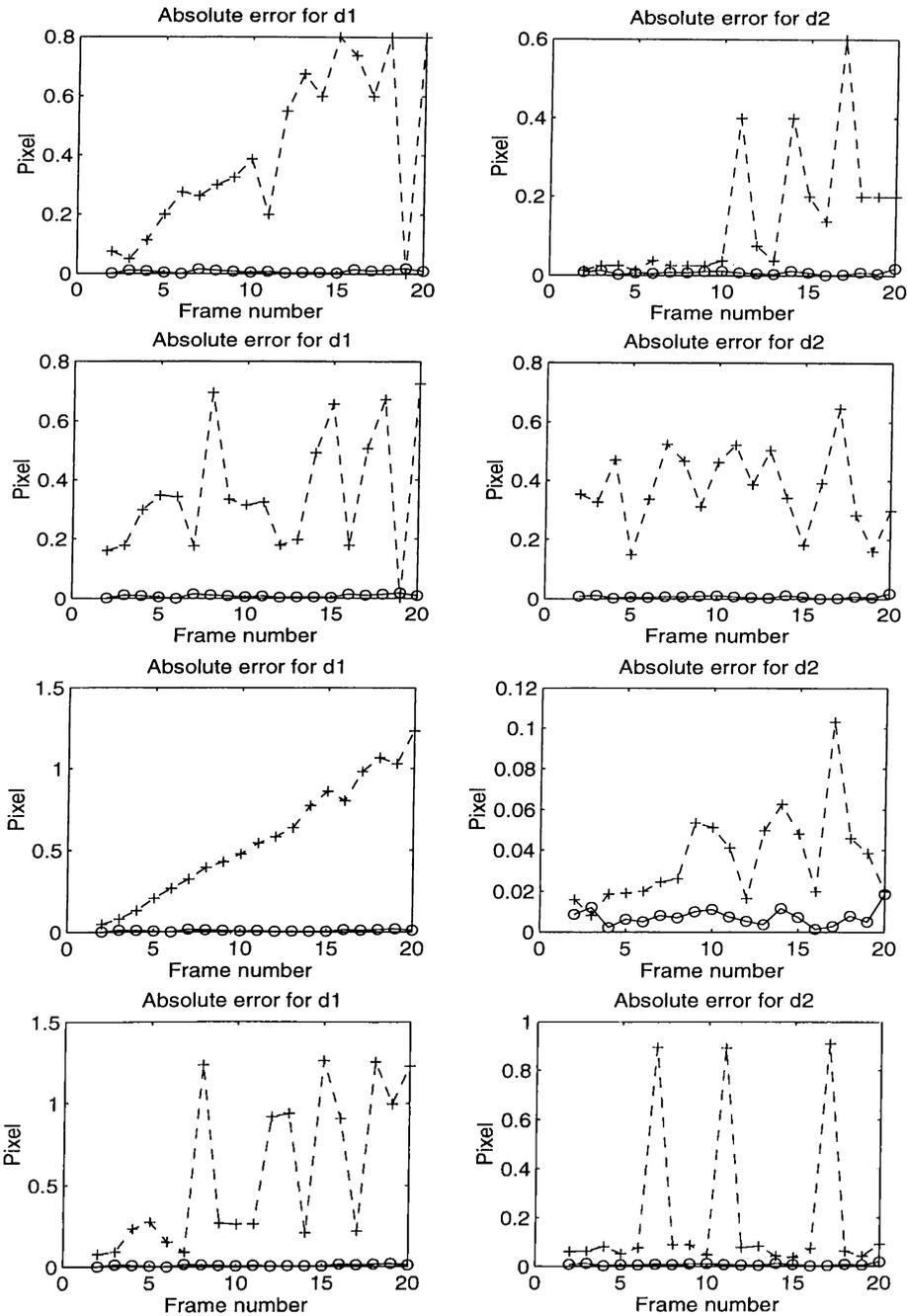


Figure 4.12:  $(\gamma_0, \eta_0) = (-0.02, -2)$ . Comparison of absolute displacement estimation errors of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for CT-2 Sequence. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares the absolute errors for  $d_1$ , while the second column compares the absolute errors for  $d_2$ .

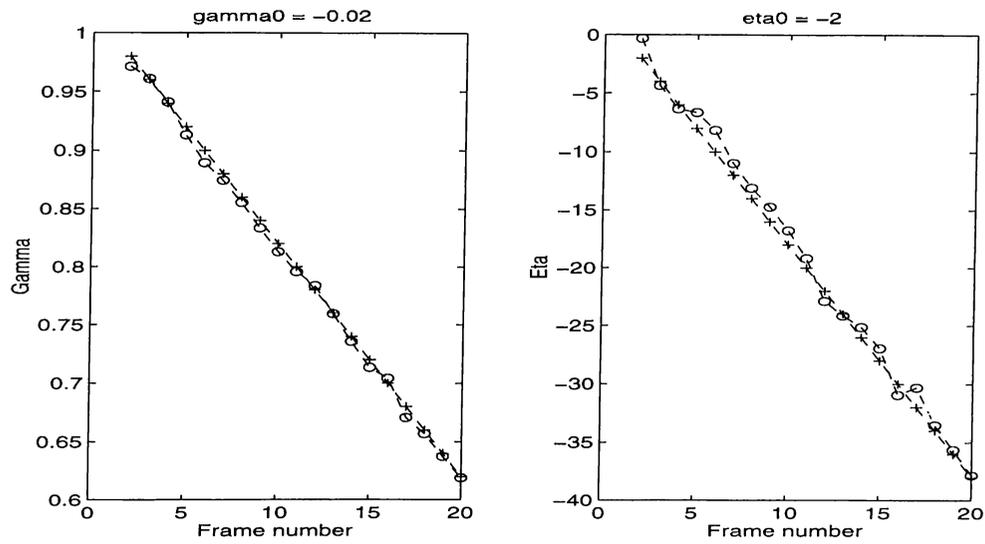


Figure 4.13: The  $\gamma$  and  $\eta$  values found for the CT-2 Sequence in which  $(\gamma_0, \eta_0) = (-0.02, -2)$ . The “\*” and “+” signs denote the estimated and true values for  $\gamma$  (on the left) and  $\eta$  (on the right) respectively.

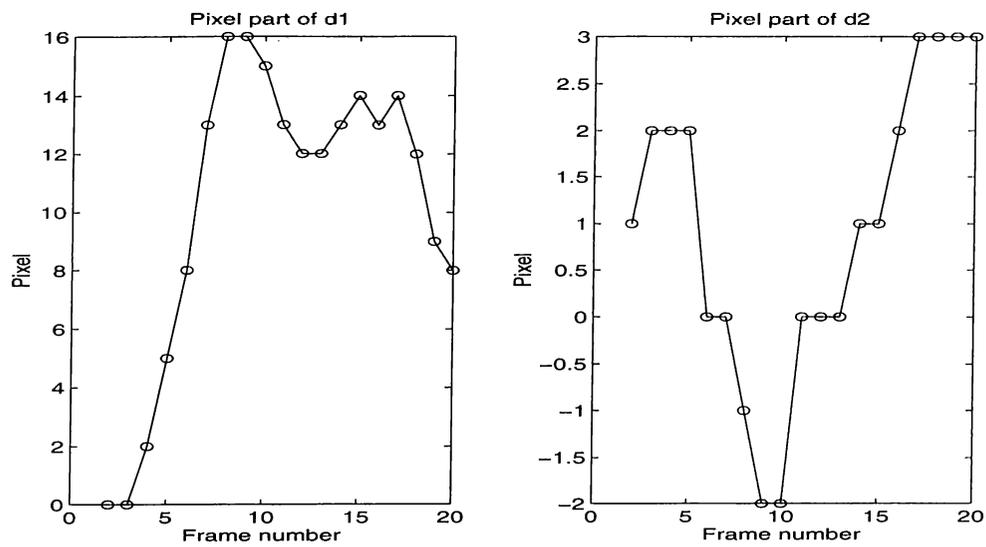


Figure 4.14: Pixel part of the displacements obtained by the phase correlation method for the “Bilkent-2” Sequence.

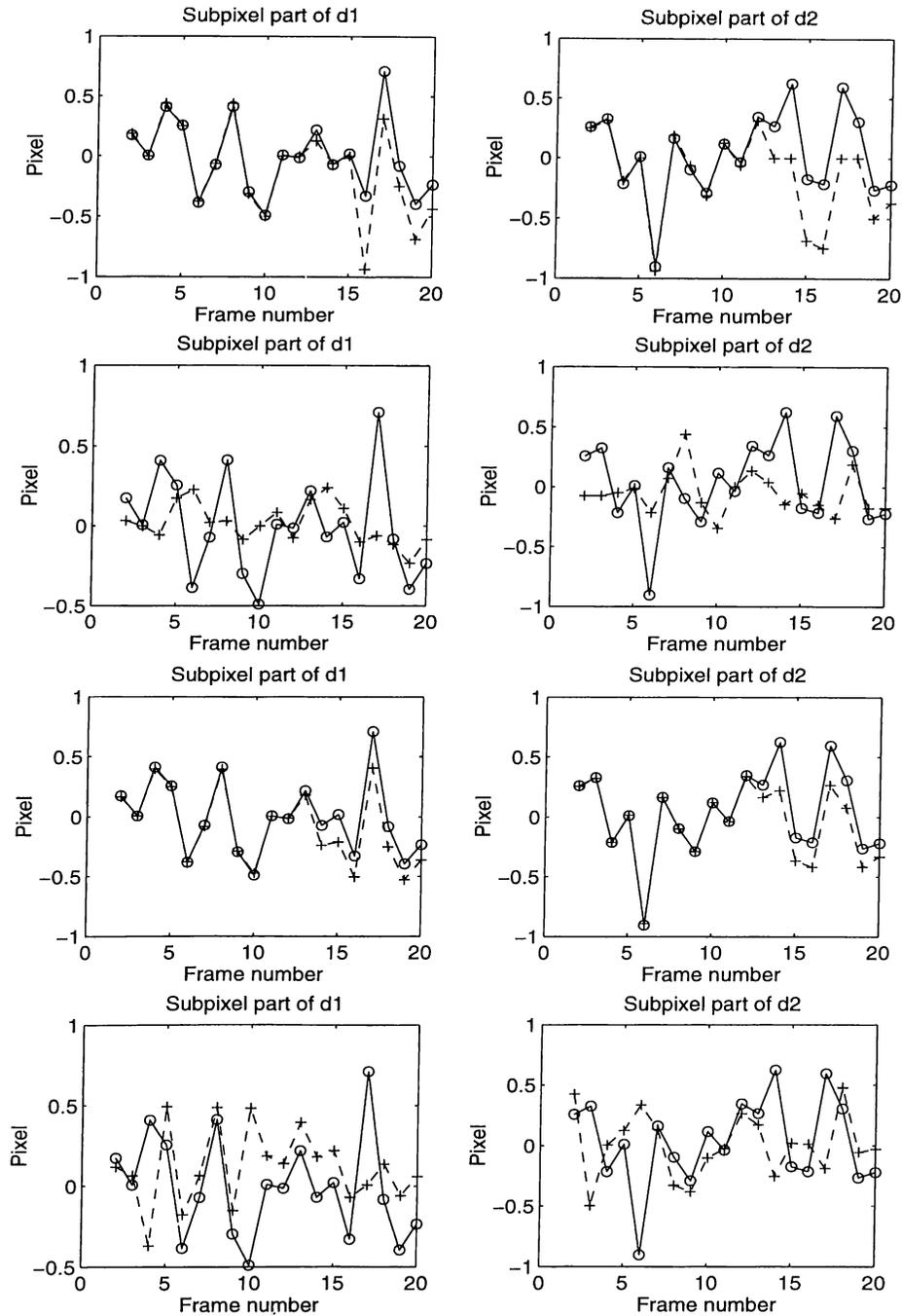


Figure 4.15: Plots of subpixel part of the displacements of the near-closed-form solution with those of the logarithmic search (first row), phase correlation interpolation (second row), differentiation (third row), and cross-correlation interpolation (fourth row) method for “Bilkent-2” Sequence. “Circles” denote the results obtained by the near-closed-form solution, while “+” signs denote the results obtained by the other methods. The first column compares  $d_1$ , while the second column compares  $d_2$ .

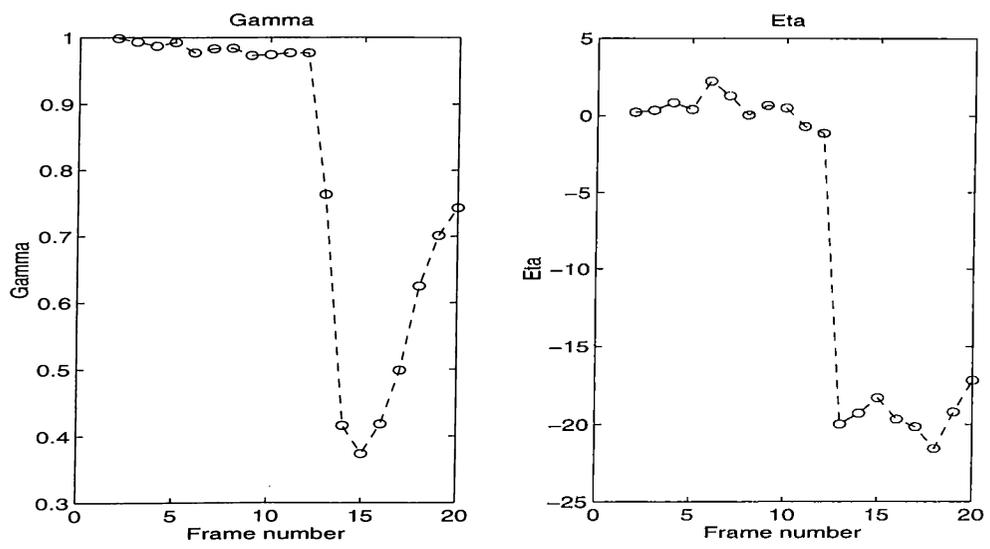


Figure 4.16: The  $\gamma$  and  $\eta$  values found for the Bilkent-2 Sequence.

## Chapter 5

# CONCLUSIONS AND FUTURE WORK

In this thesis, a comprehensive review of image registration literature is provided. The existing body of research has been analyzed in four groups which are the search-based methods, correlation methods, differentiation methods, and feature matching techniques and others. Among these four groups, correlation methods and differentiation methods provide closed-form solutions for the estimation of misregistration parameters.

Chapter 2 introduced an efficient search method for the estimation of sub-pixel displacements existing among misregistered frames. It is demonstrated that the proposed algorithm is more efficient than the traditional logarithmic and exhaustive search methods in terms of the CPU time required. Moreover, the CPU time of the proposed algorithm does not increase significantly with increasing search accuracy unlike the exhaustive search method.

In Chapter 3, a novel near-closed form solution, which is even more efficient than the method given in Chapter 2 because it does not employ any search unless it is absolutely necessary is proposed. The proposed algorithm efficiently finds a near-closed-form solution to estimation of the subpixel displacement

between two images given its pixel part. The pixel part of the displacement is found using the well-known phase correlation algorithm. In the proposed method, the criterion function to be minimized is chosen as the mean squared error between the displaced frames and the image intensities at subpixel locations are represented using bilinear interpolation. The closed-form solution is achieved by minimizing the mean squared error function analytically rather than using a search.

The proposed algorithm is compared with four popular methods for subpixel displacement estimation existing in the literature. These methods are logarithmic search, differentiation, phase correlation and cross-correlation surface interpolation techniques. The performances of the five algorithms are compared using three different unsteady image sequences. Two of them are synthetically generated (Text and CT Sequences) and one of them is a real-life sequence (Bilkent Sequence). The synthetically generated sequences contain only known random subpixel displacements and the Bilkent Sequence also contains pixel displacements which are unknown. When these sequences are displayed at a rate of 30 frames/second, the misregistration of the images causes a disturbing jitter in the scene. When they are displayed after registration with the near-closed-form solution they look steady.

If we compare the subpixel displacement estimation algorithms in detail, we can say that for the Text Sequence, the near-closed-form solution performs the best among the five methods at 20 dB and 10 dB PSNR levels. At 5 dB PSNR, performances of the near-closed-form solution and cross-correlation surface interpolation methods become close to each other as the best two algorithms. The speed of the algorithms are close to each other except the logarithmic search algorithm, which is about two times slower than the other methods. For the CT Sequence, the near-closed-form solution again gives the best results among the five methods and the differentiation method closely follows it. Since we do not know the actual displacements in the Bilkent Sequence, we can not compare the displacement estimation errors of the algorithms. Nevertheless, we have observed that the Bilkent Sequence registered with the near-closed-form solution looks stable when displayed at the actual video rate. We have also noted that the sequences do not necessarily have to be progressive in order to be registered using the near-closed-form solution. Interlaced sequences can be

registered by motion compensating even and odd fields within a frame and by registering even fields of frames with respect to the even field of the reference frame. The application of the near-closed-form solution to the de-interlacing problem also resulted in high quality still images which are obtained from interlaced frames.

Chapter 4 extends the algorithm given in Chapter 3 to account for intensity variations between the images. The intensity variations between two images are modeled by multiplicative (contrast) and additive (brightness) parameters. Simulations are done both on synthetic and real sequences which contain intensity variations. Results using various Text sequences which contain different combinations of intensity variations show that the near-closed-form solution outperforms the other algorithms except the cross-correlation interpolation method, which performs almost as good as the near-closed-form solution. However, the performance of the cross-correlation interpolation method degrades significantly for the CT Sequence whereas the proposed solution still gives good results. The near-closed-form solution is able to estimate the contrast and brightness parameters almost perfectly for the Text and CT Sequences. The performance of the proposed algorithm is also very good for the real Bilkent-2 sequence which contains real intensity variations.

Finally, we can conclude from our simulation results on various sequences that the near-closed-form solution is the only algorithm that consistently has given the best results in all of the cases under consideration.

Some possible future research directions in image registration are:

- Compensation of misregistrations in image sequences with subpixel accuracy where the misregistrations are modeled by a complex motion such as affine motion.
- Application of subpixel image registration to multi-modal registration of medical images, i.e., the registration and/or fusion of MRI and CT images of the same patient belonging to the same region of the body. It

is expected that wavelet or edge-matching based techniques might give good results for data fusion.

- Removal of the unsteadiness in a sequence while tracking the intended motion of the camera. In this way, the intended motion in the scene can be preserved while eliminating the jitter.

# APPENDIX A

## PHASE CORRELATION ALGORITHM

Let the relation between two images in the spatial domain be as follows:

$$s_1(n_1, n_2) = s_2(n_1 + d_1, n_2 + d_2), \quad n_1, n_2, d_1, d_2 \in \mathcal{Z}. \quad (\text{A.1})$$

Then, their relation in the frequency domain can be expressed as:

$$S_1(w_1, w_2) = e^{j(w_1 d_1 + w_2 d_2)} S_2(w_1, w_2). \quad (\text{A.2})$$

This means, the Fourier transforms of the two images differ only by a phase difference which is related to their relative displacement. Now, if the normalized cross-power spectrum is calculated,

$$\tilde{C}(w_1, w_2) = \frac{S_1^*(w_1, w_2) S_2(w_1, w_2)}{|S_1^*(w_1, w_2) S_2(w_1, w_2)|} = e^{-j(w_1 d_1 + w_2 d_2)}. \quad (\text{A.3})$$

the exponential term which characterizes the displacements is extracted. If we take the inverse Fourier transform of the above exponential function, we are left with the phase-correlation function,

$$\tilde{c}(n_1, n_2) = \delta(n_1 - d_1, n_2 - d_2). \quad (\text{A.4})$$

The location of the peak of this delta function gives us the displacement parameters  $(d_1, d_2)$ . A sample phase-correlation function is shown in Figure A.1 which belongs to a  $40 \times 40$  block of two images which are relatively shifted. The peak of the phase correlation has been shifted to the location  $(23, 23)$  for better visualization.

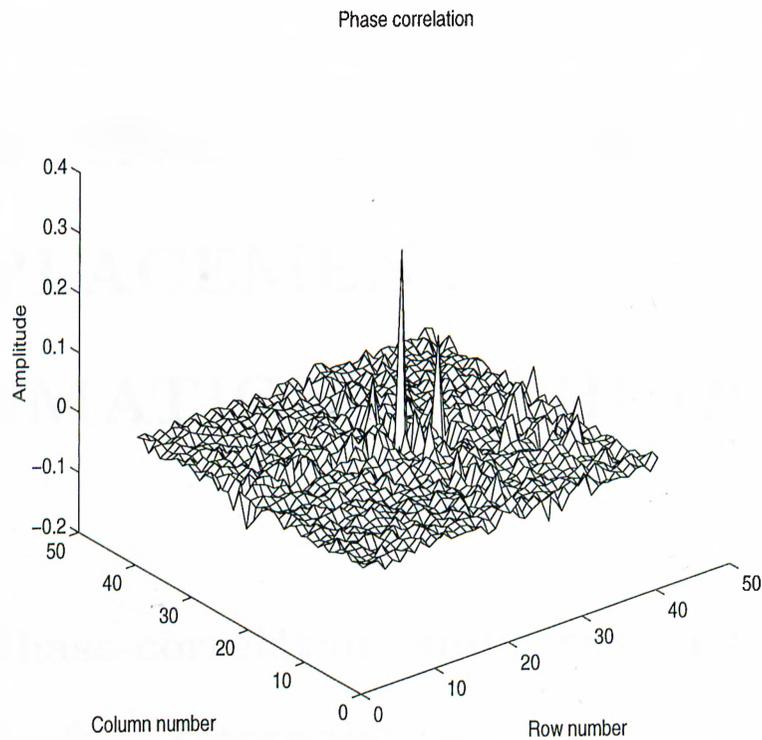


Figure A.1: The phase-correlation function for a  $40 \times 40$  block of two images which are relatively shifted.

If we do not employ normalization in (A.3) we obtain the cross-correlation function

$$\tilde{C}(w_1, w_2) = S_1^*(w_1, w_2) S_2(w_1, w_2). \quad (\text{A.5})$$

Using (A.2), it can also be written as,

$$\tilde{C}(w_1, w_2) = e^{-j(w_1 d_1 + w_2 d_2)} |S_2(w_1, w_2)|^2. \quad (\text{A.6})$$

# APPENDIX B

## EXISTING SUBPIXEL

## DISPLACEMENT

## ESTIMATION METHODS

### B.1 Phase-correlation and Cross-correlation

#### Surface Interpolation

The phase-correlation and cross-correlation algorithms can be extended to give subpixel accurate displacement vectors using phase-correlation and cross-correlation surface interpolation methods. The following approach is the same for both phase-correlation and cross-correlation interpolation methods.

Suppose the peak of the phase correlation function has been detected at a location  $(x_0, y_0)$  with an amplitude of  $c_{0,0}$  as described in Appendix A which is demonstrated in Figure B.1.

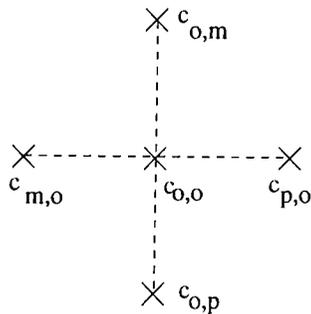


Figure B.1: The phase-correlation surface interpolation points.

The four neighbouring points around the location of the phase-correlation function peak are chosen to fit a surface. So, a quadratic surface which passes through the five points (shown in Figure B.1, with amplitudes  $c_{0,0}$ ,  $c_{0,m}$ ,  $c_{m,0}$ ,  $c_{0,p}$ , and  $c_{p,0}$ ) is to be determined. The equation of the quadratic surface to be fitted is,

$$f(x, y) = a_{xx}x^2 + a_{yy}y^2 + a_{xy}xy + a_x x + a_y y + a. \quad (\text{B.1})$$

Note that (B.1) contains six unknowns to be determined, namely  $a_{xx}$ ,  $a_{yy}$ ,  $a_{xy}$ ,  $a_x$ ,  $a_y$  and  $a$ , but we have only five points. So, one of the unknowns is chosen as a free parameter, i.e.,  $a_{xy} = 0$ . The information that we have is,

$$\begin{aligned} f(0, 0) &= c_{0,0} \\ f(1, 0) &= c_{p,0} \\ f(0, 1) &= c_{0,p} \\ f(-1, 0) &= c_{m,0} \\ f(0, -1) &= c_{0,m}. \end{aligned} \quad (\text{B.2})$$

Using (B.1) and (B.2), the following linear system is obtained,

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} a_{xx} \\ a_{yy} \\ a_x \\ a_y \\ a \end{bmatrix} = \begin{bmatrix} c_{0,0} \\ c_{p,0} \\ c_{0,p} \\ c_{m,0} \\ c_{0,m} \end{bmatrix}. \quad (\text{B.3})$$

The parameters  $a_{xx}$ ,  $a_{yy}$ ,  $a_{xy}$ ,  $a_x$ ,  $a_y$  and  $a$  are found from (B.3) as

$$\begin{bmatrix} a_{xx} \\ a_{yy} \\ a_x \\ a_y \\ a \end{bmatrix} = \begin{bmatrix} -1 & 0.5 & 0 & 0.5 & 0 \\ -1 & 0 & 0.5 & 0 & 0.5 \\ 0 & 0.5 & 0 & -0.5 & 0 \\ 0 & 0 & 0.5 & 0 & -0.5 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_{0,0} \\ c_{p,0} \\ c_{0,p} \\ c_{m,0} \\ c_{0,m} \end{bmatrix}. \quad (\text{B.4})$$

Then, to determine the subpixel location where the quadratic function (B.1) takes its maximum value, the derivatives of (B.1) with respect to  $x$  and  $y$  are found and equated to zero.i.e.,

$$\begin{aligned} \frac{\partial f(x,y)}{\partial x} &= 2a_{xx}x^* + a_x = 0 \\ \frac{\partial f(x,y)}{\partial y} &= 2a_{yy}y^* + a_y = 0. \end{aligned} \quad (\text{B.5})$$

where  $x^*$  and  $y^*$  denote the coordinates of the subpixel maximum. From (B.4) and (B.5), the subpixel displacements are found as

$$\begin{aligned} x^* &= -\frac{a_x}{2a_{xx}} \\ y^* &= -\frac{a_y}{2a_{yy}}. \end{aligned} \quad (\text{B.6})$$

## B.2 Logarithmic Search Method

In logarithmic search, a predetermined subset of subpixel displacements are searched down to the desired accuracy and the mean squared error criterion is evaluated. Figure B.2 illustrates the three-step logarithmic search. Crosses denote the actual pixel locations. In the first pass, the subpixel displacements shown with squares at 1/2 pixel positions and the center location are tested. This is done by shifting the frame to be registered by the subpixel displacement at hand using bilinear interpolation. The location which minimizes the MSE (the shaded square) is chosen as the center for the next pass. Then, for 1/4 pixel accuracy, the horizontal and vertical distances are halved and the locations

around the shaded square, shown with circles are evaluated. Let the filled circle be the one that minimizes the MSE among them. Finally, the distance is again halved and the locations shown with triangles are tested for  $1/8$  pixel accuracy. The displacement that minimizes the MSE is chosen as the actual subpixel shift (the black triangle). This process can be extended if more accuracy is desired.

Logarithmic search methods may give suboptimal results but they are much faster than the exhaustive search method. For an accuracy of  $2^{-n}$  pixels,  $9 + 8(n - 1)$  different displacements should be evaluated. This corresponds to a total of  $9N_1N_2[9 + 8(n - 1)]$  multiplications and  $5N_1N_2[9 + 8(n - 1)]$  summations. Note that this number increases approximately linearly with  $n$ .

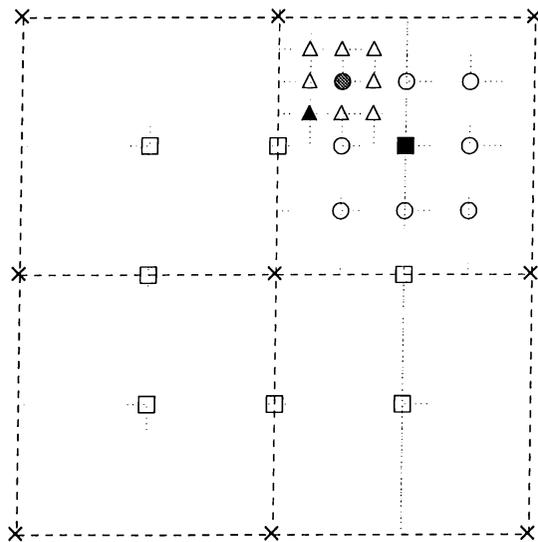


Figure B.2: Logarithmic (three-step search)

### B.3 Differentiation Method

Different approaches to the differentiation method exist in the literature [13, 29]. The method that uses the optical flow equation will be described here [29]. The optical flow equation resides on the assumption that the intensity remains constant along a motion trajectory. Let  $s(x_1, x_2, t)$  denote the continuous image at time  $t$ . Then, the above statement can be expressed as :

$$\frac{ds(x_1, x_2, t)}{dt} = 0. \quad (\text{B.7})$$

Using the chain rule of differentiation:

$$\frac{\partial s(x_1, x_2, t)}{\partial x_1} v_1(x_1, x_2, t) + \frac{\partial s(x_1, x_2, t)}{\partial x_2} v_2(x_1, x_2, t) + \frac{\partial s(x_1, x_2, t)}{\partial t} = 0. \quad (\text{B.8})$$

where  $v_1(x_1, x_2, t) = dx_1/dt$  and  $v_2(x_1, x_2, t) = dx_2/dt$  are the components of the coordinate velocity vector.

The Lucas and Kanade method [29] assumes that the motion vector is constant over a block of pixels  $\mathcal{B}$ . This assumption is suitable to our registration purposes because we are looking for a global displacement vector which is valid for the whole image; that is

$$\mathbf{v}(\mathbf{x}, t) = \mathbf{v}(t) = [v_1(t) \ v_2(t)]^T \quad \mathbf{x} \in \mathcal{B}. \quad (\text{B.9})$$

where  $\mathbf{x} = [x_1 \ x_2]$ . The error in the optical flow equation over a block of pixels is given as:

$$E = \sum_{\mathbf{x} \in \mathcal{B}} \left( \frac{\partial s(\mathbf{x}, t)}{\partial x_1} v_1(\mathbf{x}, t) + \frac{\partial s(\mathbf{x}, t)}{\partial x_2} v_2(\mathbf{x}, t) + \frac{\partial s(\mathbf{x}, t)}{\partial t} \right)^2. \quad (\text{B.10})$$

If the partial derivatives of (B.10) with respect to  $v_1(t)$  and  $v_2(t)$  are set equal to zero:

$$\begin{aligned} \sum_{\mathbf{x} \in \mathcal{B}} \left( \frac{\partial s(\mathbf{x}, t)}{\partial x_1} \hat{v}_1(t) + \frac{\partial s(\mathbf{x}, t)}{\partial x_2} \hat{v}_2(t) + \frac{\partial s(\mathbf{x}, t)}{\partial t} \right) \frac{\partial s(\mathbf{x}, t)}{\partial x_1} &= 0 \\ \sum_{\mathbf{x} \in \mathcal{B}} \left( \frac{\partial s(\mathbf{x}, t)}{\partial x_1} \hat{v}_1(t) + \frac{\partial s(\mathbf{x}, t)}{\partial x_2} \hat{v}_2(t) + \frac{\partial s(\mathbf{x}, t)}{\partial t} \right) \frac{\partial s(\mathbf{x}, t)}{\partial x_2} &= 0. \end{aligned} \quad (\text{B.11})$$

solving these equations for the estimated velocity values  $\hat{v}_1(t)$  and  $\hat{v}_2(t)$ :

$$\begin{bmatrix} \hat{v}_1(t) \\ \hat{v}_2(t) \end{bmatrix} = \begin{bmatrix} \sum_{\mathbf{x} \in \mathcal{B}} \frac{\partial s(\mathbf{x}, t)}{\partial x_1} \frac{\partial s(\mathbf{x}, t)}{\partial x_1} & \sum_{\mathbf{x} \in \mathcal{B}} \frac{\partial s(\mathbf{x}, t)}{\partial x_1} \frac{\partial s(\mathbf{x}, t)}{\partial x_2} \\ \sum_{\mathbf{x} \in \mathcal{B}} \frac{\partial s(\mathbf{x}, t)}{\partial x_1} \frac{\partial s(\mathbf{x}, t)}{\partial x_2} & \sum_{\mathbf{x} \in \mathcal{B}} \frac{\partial s(\mathbf{x}, t)}{\partial x_2} \frac{\partial s(\mathbf{x}, t)}{\partial x_2} \end{bmatrix}^{-1} \begin{bmatrix} - \sum_{\mathbf{x} \in \mathcal{B}} \frac{\partial s(\mathbf{x}, t)}{\partial x_1} \frac{\partial s(\mathbf{x}, t)}{\partial t} \\ - \sum_{\mathbf{x} \in \mathcal{B}} \frac{\partial s(\mathbf{x}, t)}{\partial x_2} \frac{\partial s(\mathbf{x}, t)}{\partial t} \end{bmatrix}. \quad (\text{B.12})$$

This gives us the displacement between two frames with subpixel accuracy.

# APPENDIX C

## FORMULAS USED IN THE NEAR-CLOSED-FORM SOLUTION

### C.1 Basic Summations

In the following, the summations are over  $n_1, n_2 \in \mathcal{B}$  and the results of the summations are normalized by  $N_1 N_2$ . Thus, for example,

$$\sum s_1(n_1, n_2) \doteq \frac{1}{N_1 N_2} \sum_{n_1, n_2 \in \mathcal{B}} s_1(n_1, n_2). \quad (\text{C.1})$$

The definition of the basic summations are now given as follows.

$$A_{0,0} = \sum s_1(n_1, n_2),$$

$$A_{0,0; 0,0} = \sum s_1^2(n_1, n_2),$$

$$B_{0,0; i,j} = \sum s_1(n_1, n_2) s_2(n_1 + i, n_2 + j), \quad i, j = -1, 0, 1,$$

$$D_{i,j} = \sum s_2(n_1 + i, n_2 + j), \quad i, j = -1, 0, 1,$$

$$D_{i,j; k,\ell} = \sum s_2(n_1 + i, n_2 + j) s_2(n_1 + k, n_2 + \ell), \quad i, j, k, \ell = -1, 0, 1, \quad (i, k), (j, \ell) \neq (\pm 1, \mp 1).$$

We note that the total number of distinct basic summations is 49  
(1+1+9+9+29).

## C.2 MSE Coefficients

In the following, we express  $C_0^{(i)}, \dots, C_8^{(i)}$ , in terms of the basic summations. The scalars  $I$  and  $J$  are determined by the quadrant number as defined in Chapter 2.

$$\begin{aligned} C_0^{(i)} &= A_{0,0; 0,0} - 2B_{0,0; 0,0} + D_{0,0; 0,0}, \\ C_1^{(i)} &= 2I(B_{0,0; 0,0} - B_{0,0; I,0} - D_{0,0; 0,0} + D_{0,0; I,0}), \\ C_2^{(i)} &= 2J(B_{0,0; 0,0} - B_{0,0; 0,J} - D_{0,0; 0,0} + D_{0,0; 0,J}), \\ C_3^{(i)} &= 2IJ(-B_{0,0; 0,0} + B_{0,0; I,0} + B_{0,0; 0,J} - B_{0,0; I,J} \\ &\quad + 2D_{0,0; 0,0} - 2D_{0,0; I,0} - 2D_{0,0; 0,J} + D_{0,0; I,J} + D_{I,0; 0,J}), \\ C_4^{(i)} &= D_{0,0; 0,0} - 2D_{0,0; I,0} + D_{I,0; I,0}, \\ C_5^{(i)} &= D_{0,0; 0,0} - 2D_{0,0; 0,J} + D_{0,J; 0,J}, \\ C_6^{(i)} &= 2J(-D_{0,0; 0,0} + 2D_{0,0; I,0} + D_{0,0; 0,J} - D_{0,0; I,J} - D_{I,0; I,0} - D_{I,0; 0,J} + D_{I,0; I,J}), \\ C_7^{(i)} &= 2I(-D_{0,0; 0,0} + D_{0,0; I,0} + 2D_{0,0; 0,J} - D_{0,0; I,J} - D_{I,0; 0,J} - D_{0,J; 0,J} + D_{0,J; I,J}), \\ C_8^{(i)} &= D_{0,0; 0,0} - 2D_{0,0; I,0} - 2D_{0,0; 0,J} + 2D_{0,0; I,J} + D_{I,0; I,0} + 2D_{I,0; 0,J} \\ &\quad - 2D_{I,0; I,J} + D_{0,J; 0,J} - 2D_{0,J; I,J} + D_{I,J; I,J}. \end{aligned}$$

### C.3 The Coefficients of the Fifth Order Polynomial

In the following, we give the expressions for  $E_0^{(i)}, \dots, E_5^{(i)}$  in terms of  $C_1^{(i)}, \dots, C_8^{(i)}$ . For notational simplicity, we omit the superscript  $(i)$  in the following equations as the expressions are the same for each  $i = 1, 2, 3, 4$ .

$$E_0 = -4C_2C_4^2 + 2C_1C_3C_4 - C_1^2C_6$$

$$E_1 = -8C_4^2C_5 + 4C_1C_4C_7 + 2C_3^2C_4 - 8C_2C_4C_6 - 2C_1^2C_8$$

$$E_2 = 2C_1C_6C_7 - 4C_2C_6^2 + C_3^2C_6 - 8C_2C_4C_8 - 16C_4C_5C_6 - 2C_1C_3C_8 + 6C_3C_4C_7$$

$$E_3 = -8C_2C_6C_8 + 4C_4C_7^2 - 16C_4C_5C_8 + 4C_3C_6C_7 - 8C_5C_6^2$$

$$E_4 = 3C_6C_7^2 - 4C_2C_8^2 - 16C_5C_6C_8 + 2C_3C_7C_8$$

$$E_5 = 2C_7^2C_8 - 8C_5C_8^2$$

### C.4 The Coefficients for Contrast and Brightness Parameters

Let

$$\Delta = \frac{1}{A_{0,0; 0,0} - A_{0,0}^2}$$

Then

$$C_0^{(i)} = \Delta[B_{0,0; 0,0} - A_{0,0}D_{0,0}]$$

$$C_1^{(i)} = I\Delta[B_{0,0; I,0} - B_{0,0; 0,0} - A_{0,0}(D_{I,0} - D_{0,0})]$$

$$G_2^{(i)} = J\Delta[B_{0,0; 0,J} - B_{0,0; 0,0} - A_{0,0}(D_{0,J} - D_{0,0})]$$

$$G_3^{(i)} = IJ\Delta[B_{0,0; 0,0} - B_{0,0; I,0} - B_{0,0; 0,J} + B_{0,0; I,J} - A_{0,0}(D_{0,0} - D_{I,0} - D_{0,J} + D_{I,J})]$$

and

$$H_0^{(i)} = \Delta[A_{0,0; 0,0}D_{0,0} - A_{0,0}B_{0,0; 0,0}]$$

$$H_1^{(i)} = I\Delta[A_{0,0; 0,0}(D_{I,0} - D_{0,0}) - A_{0,0}(B_{0,0; I,0} - B_{0,0; 0,0})]$$

$$H_2^{(i)} = J\Delta[A_{0,0; 0,0}(D_{0,J} - D_{0,0}) - A_{0,0}(B_{0,0; 0,J} - B_{0,0; 0,0})]$$

$$H_3^{(i)} = IJ\Delta[A_{0,0; 0,0}(D_{0,0} - D_{I,0} - D_{0,J} + D_{I,J}) \\ - A_{0,0}(B_{0,0; 0,0} - B_{0,0; I,0} - B_{0,0; 0,J} + B_{0,0; I,J})]$$

In the above, the scalars  $I$  and  $J$  are again determined by the quadrant number  $i$  as given in Chapter 2.

## C.5 MSE Coefficients in the Case of Intensity Variations

In the following, we give the expressions for  $C_0^{(i)}, \dots, C_8^{(i)}$ , in terms of the basic summations. The scalars  $I$  and  $J$  are determined by the quadrant number as defined in Chapter 2.

$$C_0^{(i)} = H_0^2 + 2H_0G_0A_{0,0} + G_0^2A_{0,0; 0,0} - 2G_0B_{0,0; 0,0} - 2H_0D_{0,0} + D_{0,0; 0,0}$$

$$C_1^{(i)} = 2[H_0H_1 + (H_0G_1 + H_1G_0)A_{0,0} + G_0G_1A_{0,0; 0,0} + (G_0 - G_1)B_{0,0; 0,0} - G_0B_{0,0; I,0} \\ + (H_0 - H_1)D_{0,0} - D_{0,0; 0,0} + D_{0,0; I,0} - H_0D_{I,0}]$$

$$C_2^{(i)} = 2[H_0H_2 + (H_0G_2 + H_2G_0)A_{0,0} + G_0G_2A_{0,0; 0,0} + (G_0 - G_2)B_{0,0; 0,0} - G_0B_{0,0; 0,J} \\ + (H_0 - H_2)D_{0,0} - D_{0,0; 0,0} + D_{0,0; 0,J} - H_0D_{0,J}]$$

$$C_3^{(i)} = 2[H_1H_2 + H_0H_3 + (H_1G_2 + H_2G_1 + H_0G_3 + H_3G_0)A_{0,0} + (G_1G_2 + G_0G_3)A_{0,0; 0,0}$$

$$\begin{aligned}
& +(G_1 + G_2 - G_3 - G_0)B_{0,0; 0,0} + (G_0 - G_2)B_{0,0; I,0} \\
& +(G_0 - G_1)B_{0,0; 0,J} - G_0B_{0,0; I,J} \\
& +(H_1 + H_2 - H_3 - H_0)D_{0,0} + 2D_{0,0; 0,0} - 2D_{0,0; I,0} - 2D_{0,0; 0,J} + D_{0,0; I,J} \\
& +(H_0 - H_2)D_{I,0} + D_{I,0; 0,J} + (H_0 - H_1)D_{0,J} - H_0D_{I,J}] \\
C_4^{(i)} &= H_1^2 + 2H_1G_1A_{0,0} + G_1^2A_{0,0; 0,0} + 2G_1B_{0,0; 0,0} - 2G_1B_{0,0; I,0} \\
& + 2H_1D_{0,0} + D_{0,0; 0,0} - 2D_{0,0; I,0} - 2H_1D_{I,0} + D_{I,0; I,0} \\
C_5^{(i)} &= H_2^2 + 2H_2G_2A_{0,0} + G_2^2A_{0,0; 0,0} + 2G_2B_{0,0; 0,0} - 2G_2B_{0,0; 0,J} \\
& + 2H_2D_{0,0} + D_{0,0; 0,0} - 2D_{0,0; 0,J} - 2H_2D_{0,J} + D_{0,J; 0,J} \\
C_6^{(i)} &= 2[H_1H_3 + (H_3G_1 + H_1G_3)A_{0,0} + G_1G_3A_{0,0; 0,0} \\
& - (G_1 - G_3)B_{0,0; 0,0} + (G_1 - G_3)B_{0,0; I,0} \\
& + G_1B_{0,0; 0,J} - G_1B_{0,0; I,J} - (H_1 - H_3)D_{0,0} \\
& - D_{0,0; 0,0} + 2D_{0,0; I,0} + D_{0,0; 0,J} \\
& - D_{0,0; I,J} + (H_1 - H_3)D_{I,0} - D_{I,0; I,0} \\
& - D_{I,0; 0,J} + D_{I,0; I,J} + H_1D_{0,J} - H_1D_{I,J}] \\
C_7^{(i)} &= 2[H_2H_3 + (H_3G_2 + H_2G_3)A_{0,0} + G_2G_3A_{0,0; 0,0} \\
& - (G_2 - G_3)B_{0,0; 0,0} + (G_2 - G_3)B_{0,0; 0,J} \\
& + G_2B_{0,0; I,0} - G_2B_{0,0; I,J} - (H_2 - H_3)D_{0,0} \\
& - D_{0,0; 0,0} + 2D_{0,0; 0,J} + D_{0,0; I,0} \\
& - D_{0,0; I,J} + (H_2 - H_3)D_{0,J} - D_{0,J; 0,J} \\
& - D_{0,J; I,0} + D_{0,J; I,J} + H_2D_{I,0} - H_2D_{I,J}] \\
C_8^{(i)} &= H_3^2 + 2H_3G_3A_{0,0} + G_3^2A_{0,0; 0,0} - 2G_3B_{0,0; 0,0} \\
& + 2G_3B_{0,0; I,0} + 2G_3B_{0,0; 0,J} - 2G_3B_{0,0; I,J} \\
& - 2H_3D_{0,0} + D_{0,0; 0,0} - 2D_{0,0; I,0} \\
& - 2D_{0,0; 0,J} + 2D_{0,0; I,J} + 2H_3D_{I,0} + D_{I,0; I,0} \\
& + 2D_{I,0; 0,J} - 2D_{I,0; I,J}
\end{aligned}$$

$$+2H_3D_{0,J} - 2H_3D_{I,J} + D_{0,J; 0,J} - 2D_{0,J; I,J} + D_{I,J; I,J}$$

## C.6 The Coefficients for the Brightness Parameter

$$H_0^{(i)} = D_{0,0} - A_{0,0}$$

$$H_1^{(i)} = I(D_{I,0} - D_{0,0})$$

$$H_2^{(i)} = J(D_{0,J} - D_{0,0})$$

$$H_3^{(i)} = IJ(D_{0,0} - D_{I,0} - D_{0,J} + D_{I,J})$$

In the above, the scalars  $I$  and  $J$  are again determined by the quadrant number  $i$  as given in Chapter 2.

## C.7 The Coefficients for the Contrast Parameter

$$G_0^{(i)} = B_{0,0; 0,0}/A_{0,0; 0,0}$$

$$G_1^{(i)} = I(B_{0,0; I,0} - B_{0,0; 0,0})/A_{0,0; 0,0}$$

$$G_2^{(i)} = J(B_{0,0; 0,J} - B_{0,0; 0,0})/A_{0,0; 0,0}$$

$$G_3^{(i)} = IJ(B_{0,0; 0,0} - B_{0,0; I,0} - B_{0,0; 0,J} + B_{0,0; I,J})/A_{0,0; 0,0}$$

In the above, the scalars  $I$  and  $J$  are again determined by the quadrant number  $i$  as given in Chapter 2.

## REFERENCES

- [1] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):326–376, 1992.
- [2] George Woolberg. *Digital Image Warping*. IEEE Computer Society Press, 1990.
- [3] C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. In *Proceedings of the IEEE 1975 International Conference on Cybernetics Society*, pages 163–165, 1975.
- [4] V. R. Mandava, J.M. Fitzpatrick, and D. R. Pickens. Adaptive search space scaling in digital image registration. *IEEE Trans. on Medical Imaging*, 8(3):251–262, 1989.
- [5] J. Y. Chiang and B. J. Sullivan. Coincident bit counting - a new criterion for image registration. *IEEE Trans. on Medical Imaging*, 12(1):30–38, 1993.
- [6] T. M. Buzug and J. Weese. Improving DSA images with an automatic algorithm based on template matching and an entropy measure. In *Proc. Int. Conf. on Computer Assisted Radiology*, pages 145–157, 1996.

- [7] A. Venot, J. F. Lebruchec, and J. C. Roucayrol. A new class of similarity measures for robust image registration. *Computer Vision, Graphics, and Image Processing*, 28:176–184, 1984.
- [8] P. A. Kenny, D. J. Dowsett, D. Vernon, and J. T. Ennis. A technique for digital image registration used prior to subtraction of lung images in nuclear medicine. *Physics in Medicine and Biology*, 35(5):679–685, 1990.
- [9] J. K. Cheng and T. S. Huang. Image registration by matching relational structures. *Pattern Recognition*, 17(1):149–159, 1984.
- [10] G. Healey. Hierarchical segmentation based approach to motion analysis. *Image and Vision Computing*, 11(9):570–576, 1993.
- [11] A. D. Ventura, A. Rampini, and R. Schettini. Image registration by recognition of corresponding structures. *IEEE Trans. Geoscience and Remote Sensing*, 28(3):305–314, 1990.
- [12] M. Roux. Automatic registration of spot images and digitized maps. In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pages 625–628, 1996.
- [13] Q. Tian and M. N. Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics, and Image Processing*, 35:220–233, 1986.
- [14] T. Wakahara. An iterative image registration technique using local affine transformation. *Systems and Computers in Japan*, 21(12), 1990.
- [15] H. Li, B. S. Manjunath, and S. K. Mitra. Multi-sensor image fusion using the wavelet transform. *Graphical Models and Image Processing*, 57:235–245, 1995.

- [16] M. Herbin, A. Venot, J. Y. Devaux, E. Walter, J. F. Lebruchec, L. Dubertret, and J. C. Roucayrol. Automated registration of dissimilar images: application to medical imagery. *Computer Vision, Graphics, and Image Processing*, 47:77–88, 1989.
- [17] P. Gerlot-Chiron and Y. Bizais. Registration of multimodality medical images using a region overlap criterion. *Computer Vision, Graphics, and Image Processing*, 54(5):396–406, 1992.
- [18] R. P. Woods, J. C. Mazziotta, and S. R. Cherry. MRI-PET registration with automated algorithm. *Journal of Computer Assisted Tomography*, 17(4):536–546, 1993.
- [19] R. T. Malison, E. G. Miller, R. Greene, G. McCarthy, D. S. Charney, and R. B. Innis. Computer-assisted coregistration of multislice SPECT and MR brain images by fixed external fiducials. *Journal of Computer Assisted Tomography*, 17(6):952–960, 1993.
- [20] A. M. Scott et.al. Clinical validation of SPECT and CT/MRI image registration in radiolabeled monoclonal antibody studies of colorectal carcinoma. *Journal of Nuclear Medicine*, 35(12):1976–1984, 1994.
- [21] U. Pietrzyk et.al. An interactive technique for three-dimensional image registration: Validation for PET, SPECT, MRI and CT brain studies. *Journal of Nuclear Medicine*, 35(12):2011–2018, 1994.
- [22] A. Simper. Correcting general band-to-band misregistrations. In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pages 597–601, 1996.

- [23] P. Thevanaz, U. E. Ruttimann, and M. Unser. Iterative multi-scale registration without landmarks. In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pages 228–231, 1995.
- [24] M. Irani and S. Peleg. Improving resolution by image registration. *Computer Vision, Graphics, and Image Processing*, 53(3):231–239, 1991.
- [25] A. J. Patti. *Digital video filtering for standards conversion and resolution enhancement*. PhD thesis, University of Rochester, School of Engineering, 1995.
- [26] J. P. Thiran, B. Macq, and C. Michel. Morphological registration of 3D Medical images. In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pages 253–256, 1996.
- [27] R. P. Woods, S. R. Cherry, and J. C. Mazziotta. Rapid automated algorithm for aligning and reslicing pet images. *Journal of Computer Assisted Tomography*, 16(4):620–633, 1992.
- [28] Y. L. Guilloux. A matching algorithm for horizontal motion, application to tracking . In *Proceedings of the 8th Int. IEEE Conf. on Pattern Recognition*, pages 1190–1192, Paris, October 1986.
- [29] A. Murat Tekalp. *Digital Video Processing*. Prentice-Hall, New Jersey, 1995.
- [30] D. R. Sullivan and J. S. Martin. A coarse correlation tracker for image registration. *IEEE Trans. Aerospace and Electronic Systems*, AES-17(1):29–34, 1981.

- [31] A. Venot, L. Pronzato, and E. Walter. Comments about the coincident bit counting (CBC) criterion for image registration. *IEEE Trans. on Medical Imaging*, 13(3):565–566, 1994.
- [32] J. J. Pearson, D. C. Hines, S. Golosman, and C. D. Kuglin. Video rate image correlation processor. *Proc. SPIE, (Application of Digital Image Processing, IOCC)*, 119:197–205, 1977.
- [33] R. Manduchi and G. A. Mian. Accuracy analysis for correlation-based image registration algorithms. *Proc. IEEE Int. Symp. Circuits and Systems*, 1:834–837, 1993.
- [34] S. Alliney and C. Morandi. Digital image registration using projections. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, PAMI-8(2):222–233, 1986.
- [35] C. Morandi, F. Piazza, and R. Capancioni. Robust 1-D equivalent of phase-correlation image registration algorithm. *Electronics Letters*, 22(7):386–388, 1986.
- [36] E. D. Castro and C. Morandi. Registration of translated and rotated images using finite fourier transforms. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, PAMI-9(5):700–703, 1987.
- [37] B. S. Srinivasa and B. N. Chatterj. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Trans. Image Process.*, 5(8):1266–1271, 1996.
- [38] D. Lee, T. F. Krile, and S. Mitra. Power cepstrum and spectrum techniques applied to image registration. *Appl. Opt.*, 27(6):1099–1106, 1988.

- [39] D. C. Barber. Registration of low resolution medical images. *Physics in Medicine and Biology*, 37(7):1485–1489, 1992.
- [40] J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg. A three-frame algorithm for estimating two-component image motion. *IEEE Trans. Pattern Analysis, and Machine Intelligence*, 14(9):886–896, 1992.
- [41] J. K. Aggarwal, L.S Davis, and W. N. Martin. Correspondence processes in dynamic scene analysis. *Proc. IEEE*, 69(5):562–572, 1981.
- [42] N. M. Alpert, J. F. Bradshaw, D. Kennedy, and J. A. Correia. The principal axes transformation - a method for image registration. *Journal of Nuclear Medicine*, 31(10):1717–1722, 1990.
- [43] L. G. Shapiro and R. M. Haralick. *Matching relational structures using discrete relaxation*. World Scientific, Teaneck, NJ, 1990.
- [44] A. Goshtasby, G. C. Stockman, and C. V. Page. A region-based approach to digital image registration with subpixel accuracy. *IEEE Trans. Geoscience and Remote Sensing*, GE-24(3):390–399, 1986.
- [45] S.D.Conte and Carl De Boor. *Elementary Numerical Anlalysis-An Algorithmic Approach*. McGraw-Hil, 1980.