Fast algorithm for subpixel-accuracy image stabilization for digital film and video

Cigdem Eroglu
A. Tanju Erdem

A Fast Algorithm for Subpixel Accuracy Image Stabilization for Digital Film and Video

Çiğdem Eroğlu[a] and A. Tanju Erdem[b]

[a]Department of Electrical and Electronics Engineering
Bilkent University, Ankara, TR-06533, Turkey

[b]Imaging Research & Advanced Development
Eastman Kodak Company, Rochester, NY 14650-1816, USA

ABSTRACT

This paper introduces a novel method for subpixel accuracy stabilization of unsteady digital films and video sequences. The proposed method offers a near-closed-form solution to the estimation of the global subpixel displacement between two frames, that causes the misregistration of them. 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. The proposed algorithm is both faster and more accurate than the search-based solutions found in the literature. Experimental results demonstrate the superiority of the proposed method to the spatio-temporal differentiation and surface fitting algorithms, as well. Furthermore, the proposed algorithm is designed so that it is insensitive to frame-to-frame intensity variations. It is also possible to estimate any affine motion between two frames by applying the proposed algorithm on three non-collinear points in the unsteady frame.

Keywords: Unsteadiness correction, image registration, motion estimation

1. INTRODUCTION

Image unsteadiness in a video or a film sequence may be caused by any unwanted or unpredictable relative movements of a camera and a scene during the recording of the scene, or that of a scanner and a motion picture film during the digitization of the film. In such applications, image stabilization problem refers to finding the global motion of each frame in the sequence with respect to a reference frame, and then correcting for each frame with the found motion parameters. In this paper, we propose an algorithm for estimating the global translational motion, i.e., the displacement, between an unsteady frame and a reference frame. We also apply the proposed algorithm to estimation of affine motion parameters between two frames. An affine motion has six parameters and it may be composed of rotation, translation, zoom and shear transformations.

The displacement (translational motion) causing the unsteadiness will, in general, have 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 Ref. 1, such as the phase correlation technique.[2] In this paper, we are interested in estimating the subpixel part of the displacement given its pixel part. It is indeed necessary to estimate the displacements down to subpixel accuracy, because subpixel translations in a sequence may cause a disturbing jitter, especially in stationary scenes.

Subpixel image registration techniques for translational motion are discussed in Ref. 3, and can be broadly classified as those that are based on intensity matching, those that employ spatio-temporal differentiation,[3,4] and those that fit a parametric surface to the cross correlation or phase correlation functions.[3,2]

Although the differentiation and surface fitting approaches to subpixel motion estimation result in closed-form solutions, the intensity matching algorithms proposed in the literature offer only search-based solutions. Both the exhaustive and logarithmic type search based solutions require significantly more computational time than the closed-form solutions, and the computational time increases with the increased subpixel accuracy. Furthermore, logarithmic

Send correspondence to Ç. Eroğlu.
Ç. E.: Email: eroglu@ee.bilkent.edu.tr; Telephone: (90-312) 266-4307; Fax: (90-312) 266-4126.
A. T. E.: Email: erdem@kodak.com; Telephone: (716) 588-0371; Fax: (716) 722-0160.

search may result in sub-optimal results. In order to eliminate these drawbacks, we propose in this paper, a near-closed-form solution to estimation of the global subpixel displacement between two frames. An extension of the method that is insensitive to intensity variations between frames, i.e., illumination effects, is also proposed. The method proposed in this paper is superior in quality to the search based solutions while it is as fast as the non-search based techniques because it uses a near-closed-form solution.

2. PROBLEM FORMULATION

Let $s_1()$ denote the displaced frame and $s_2()$ denote the unsteady frame after having been corrected for any integer pixel displacement, say, by using the phase correlation method. 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.,

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

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,

$$\begin{aligned}\tilde{s}_2(n_1 + d_1, n_2 + d_2) = s_2(n_1, n_2)(1 - d_1)(1 - d_2) \\ +s_2(n_1 + 1, n_2)(d_1)(1 - d_2) \\ +s_2(n_1, n_2 + 1)(1 - d_1)(d_2) \\ +s_2(n_1 + 1, n_2 + 1)(d_1)(d_2).\end{aligned} \tag{2}$$

We can rewrite (2) for all $-1 < d_1, d_2 < 1$, as

$$\tilde{s}_2(n_1 + d_1, n_2 + d_2) = S_0^{(i)} + S_1^{(i)} d_1 + S_2^{(i)} d_2 + S_3^{(i)} d_1 d_2, \quad (d_1, d_2) \in \mathcal{Q}^{(i)}, \tag{3}$$

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

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

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); that are defined as

$$\begin{aligned}S_0^{(i)} &= s_2(n_1, n_2) \\ S_1^{(i)} &= I[s_2(n_1 + I, n_2) - s_2(n_1, n_2)] \\ S_2^{(i)} &= J[s_2(n_1, n_2 + J) - s_2(n_1, n_2)] \\ S_3^{(i)} &= IJ[s_2(n_1 + I, n_2 + J) - s_2(n_1 + I, n_2) - s_2(n_1, n_2 + J) + s_2(n_1, n_2)],\end{aligned} \tag{5}$$

where

$$I = \begin{cases} 1 & \text{for } i = 1, 2 \\ -1 & \text{for } i = 3, 4 \end{cases}, \quad \text{and} \quad J = \begin{cases} 1 & \text{for } i = 1, 3 \\ -1 & \text{for } i = 2, 4 \end{cases}. \tag{6}$$

We define the intensity matching criterion, i.e., the mean-squared error (MSE) function for all $-1 \leq d_1, d_2 \leq 1$ as,

$$\text{MSE}^{(i)} = \frac{1}{N_1 N_2} \sum_{n_1, n_2 \in \mathcal{B}} [s_1(n_1, n_2) - \tilde{s}_2(n_1 + d_1, n_2 + d_2)]^2, \quad (d_1, d_2) \in \mathcal{Q}^{(i)}, \tag{7}$$

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) 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.

A straightforward approach to minimizing (7) would be to uniformly sample the set $\{(d_1, d_2) : -1 \leq d_1, d_2 \leq 1\}$ at a desired accuracy, compute the MSE given in (7) 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. If an accuracy of 2^{-n} pixels is desired, the exhaustive search requires the evaluation of (7) for $(2^{n+1}-1)^2$ different values of (d_1, d_2) pairs. This corresponds to $N_1 N_2$ bilinear interpolations for each (d_1, d_2) pair, which results in a total of $9N_1N_2(2^{n+1}-1)^2$ multiplications and $6N_1N_2(2^{n+1}-1)^2$ 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, which requires $9N_1N_2[9+8(n-1)]$ multiplications and $6N_1N_2[9+8(n-1)]$ summations for an accuracy of 2^{-n} pixels. 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 need to be carried out from the beginning. In the next section, a near-closed-form solution is proposed, which eliminates this need.

3. A NEAR-CLOSED-FORM SOLUTION

From (7) and (3) we obtain the following expression for MSE$^{(i)}$ in terms of the subpixel shifts d_1 and d_2:

$$\begin{aligned} \mathrm{MSE}^{(i)} &= C_0^{(i)} + C_1^{(i)} d_1 + C_2^{(i)} d_2 + C_3^{(i)} d_1 d_2 + C_4^{(i)} d_1^2 + C_5^{(i)} d_2^2 \\ &\quad + C_6^{(i)} d_1^2 d_2 + C_7^{(i)} d_1 d_2^2 + C_8^{(i)} d_1^2 d_2^2, \quad i = 1, 2, 3, 4, \end{aligned} \tag{8}$$

where (i) denotes one of the four quadrants in the Cartesian coordinates as defined in Sect. 1, and the coefficients $C_0^{(i)}, \ldots, C_8^{(i)}$ are computed over the two images using the basic summations as described in APPENDIX A and APPENDIX B.

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

$$\frac{\partial \mathrm{MSE}^{(i)}}{\partial d_1} = C_1^{(i)} + C_3^{(i)} d_2 + 2C_4^{(i)} d_1 + 2C_6^{(i)} d_1 d_2 + C_7^{(i)} d_2^2 + 2C_8^{(i)} d_1 d_2^2 = 0 \tag{9}$$

$$\frac{\partial \mathrm{MSE}^{(i)}}{\partial d_2} = C_2^{(i)} + C_3^{(i)} d_1 + 2C_5^{(i)} d_2 + C_6^{(i)} d_1^2 + 2C_7^{(i)} d_1 d_2 + 2C_8^{(i)} d_1^2 d_2 = 0. \tag{10}$$

We note that the equation (9) is linear in d_1. Thus we can express d_1 as a function of d_2 as

$$d_1 = -0.5 \frac{C_1^{(i)} + C_3^{(i)} d_2 + C_7^{(i)} d_2^2}{C_4^{(i)} + C_6^{(i)} d_2 + C_8^{(i)} d_2^2}. \tag{11}$$

Then, we substitute (11) in the equation (10), to obtain the following polynomial equation in d_2:

$$E_5 d_2^5 + E_4 d_2^4 + E_3 d_2^3 + E_2 d_2^2 + E_1 d_2 + E_0 = 0, \tag{12}$$

where the coefficients $E_0, \ldots, E_5$, are defined in terms of $C_0, \ldots, C_8$. The definitions of $E_0, \ldots, E_5$ are given in APPENDIX C. Unfortunately, there does not exist an algebraic formula for the zeros of a fifth degree polynomial. Thus, the zeros of (12) are obtained numerically using the Muller's method.[5] Once the solution for d_2 is obtained, d_1 is calculated from (11).

Since (12) is a fifth degree polynomial, for each quadrant $Q^{(i)}$, 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 $Q^{(i)}$, only the solutions (d_1, d_2) that are in $Q^{(i)}$ 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 an efficient exhaustive search method which uses (8) instead of (7) 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 A over a specified block of pixels. Note that only 39 basic summations are computed at this step.
2. Compute the MSE coefficients $C_0^{(i)}, \cdots, C_8^{(i)}$, given in APPENDIX B for each quadrant, i.e., for each $i = 1, 2, 3, 4$.
3. Compute the coefficients $E_0, \cdots, E_5$, of the fifth degree polynomial as given in APPENDIX C for each quadrant.
4. Find the zeroes of (12) 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 (8) using an efficient exhaustive search method.[6]

4. ACCOUNTING FOR INTENSITY VARIATIONS

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

$$I_c = \gamma I_r + \eta, \tag{13}$$

where γ and η 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 (7) as

$$\mathrm{MSE}^{(i)} = \frac{1}{N_1 N_2} \sum_{n_1, n_2 \in \mathcal{B}} [\gamma s_1(n_1, n_2) + \eta - \tilde{s}_2(n_1 + d_1, n_2 + d_2)]^2, \quad (d_1, d_2) \in \mathcal{Q}^{(i)}. \tag{14}$$

We note that in addition to d_1 and d_2, two new parameters, namely γ and η, need to be determined for each frame. We use the approach suggested in Ref. 7 to first find the optimal solution for γ and η in terms of d_1 and d_2 by setting $\partial\mathrm{MSE}/\partial\gamma = 0$ and $\partial\mathrm{MSE}/\partial\eta = 0$. That is,

$$\begin{aligned} \frac{\partial \mathrm{MSE}^{(i)}}{\partial \gamma} &= \sum_{n_1, n_2 \in \mathcal{B}} [\gamma s_1(n_1, n_2) + \eta - \tilde{s}_2(n_1 + d_1, n_2 + d_2)] s_1(n_1, n_2) = 0, \\ \frac{\partial \mathrm{MSE}^{(i)}}{\partial \eta} &= \sum_{n_1, n_2 \in \mathcal{B}} [\gamma s_1(n_1, n_2) + \eta - \tilde{s}_2(n_1 + d_1, n_2 + d_2)] = 0. \end{aligned} \tag{15}$$

Note that these equations are linear in γ and η:

$$\begin{bmatrix} \sum s_1^2(n_1, n_2) & \sum s_1(n_1, n_2) \\ \sum s_1(n_1, n_2) & N_1 N_2 \end{bmatrix} \begin{bmatrix} \gamma \\ \eta \end{bmatrix} = \begin{bmatrix} \sum s_1(n_1, n_2) \tilde{s}_2(n_1 + d_1, n_2 + d_2) \\ \sum \tilde{s}_2(n_1 + d_1, n_2 + d_2) \end{bmatrix}. \tag{16}$$

If we substitute the bilinear interpolation expression given in (3) for $\tilde{s}_2(n_1 + d_1, n_2 + d_2)$ in (16), then the solution (γ^*, η^*) to (16) becomes a function of d_1 and d_2 in the following form

$$\begin{aligned} \gamma^* &= G_0^{(i)} + G_1^{(i)} d_1 + G_2^{(i)} d_2 + G_3^{(i)} d_1 d_2, \\ \eta^* &= H_0^{(i)} + H_1^{(i)} d_1 + H_2^{(i)} d_2 + H_3^{(i)} d_1 d_2, \end{aligned} \tag{17}$$

where the coefficients $G_0^{(i)}, \ldots, H_3^{(i)}$ are given in terms of the basic summations that are defined in APPENDIX A. The actual expressions for $G_0^{(i)}, \ldots, H_3^{(i)}$ in terms of the basic summations are provided in APPENDIX D. We note that the optimal values γ^* and η^* given in (17) are bilinear in d_1 and d_2. Thus, when (3) and the expression for (γ^*, η^*) are substituted in (14), the expression within the square brackets will still be bilinear in d_1 and d_2:

$$\begin{aligned} & G_0^{(i)} s_1(n_1, n_2) + H_0^{(i)} - S_0^{(i)} + [G_1^{(i)} s_1(n_1, n_2) + H_1^{(i)} - S_1^{(i)}] d_1 \\ & + [G_2^{(i)} s_1(n_1, n_2) + H_2^{(i)} - S_2^{(i)}] d_2 + [G_3^{(i)} s_1(n_1, n_2) + H_3^{(i)} - S_3^{(i)}] d_1 d_2. \end{aligned} \tag{18}$$

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 (8):

$$\begin{aligned} \text{MSE}^{(i)} &= C_0^{(i)} + C_1^{(i)} d_1 + C_2^{(i)} d_2 + C_3^{(i)} d_1 d_2 + C_4^{(i)} d_1^2 + C_5^{(i)} d_2^2 \\ &\quad + C_6^{(i)} d_1^2 d_2 + C_7^{(i)} d_1 d_2^2 + C_8^{(i)} d_1^2 d_2^2, \quad i = 1, 2, 3, 4. \end{aligned} \tag{19}$$

Of course, the expressions for the coefficients $C_0^{(i)}, \ldots, C_8^{(i)}$, in terms of the basic summations will now be different than those used in Section 3. The new expressions for the coefficients $C_0^{(i)}, \ldots, C_8^{(i)}$, are given in APPENDIX E.

In order to minimize (8) with respect to d_1 and d_2, we again solve $\partial\text{MSE}/\partial d_1 = 0$ and $\partial\text{MSE}/\partial d_2 = 0$ simultaneously. Carrying out the steps given in Equations (9) and (10), we get the same expression for d_1 in terms of d_2 as given in (11). The expressions for the coefficients of the fifth degree polynomial (12) in terms of new coefficients $C_0^{(i)}, \ldots, C_8^{(i)}$ will still be as given in APPENDIX C.

Thus, the incorporation of intensity variations into the near-closed-form solution is achieved by simply re-defining the coefficients $C_0^{(i)}, \ldots, C_8^{(i)}$, 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 Section 3 is only in the computation of the coefficients $C_0^{(i)}, \ldots, C_8^{(i)}$; otherwise the two algorithms are exactly the same.

In particular, if frame-to-frame intensity variations are caused only by the changes in the brightness parameter (i.e., the contrast parameter is unchanged), then the MSE expression in (14) simplifies to

$$\text{MSE}^{(i)} = \frac{1}{N_1 N_2} \sum_{n_1, n_2 \in \mathcal{B}} [s_1(n_1, n_2) + \eta - \tilde{s}_2(n_1 + d_1, n_2 + d_2)]^2, \quad (d_1, d_2) \in \mathcal{Q}^{(i)}. \tag{20}$$

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

$$\eta^* = \frac{1}{N_1 N_2} \sum \tilde{s}_2(n_1 + d_1, n_2 + d_2) - \frac{1}{N_1 N_2} \sum s_1(n_1, n_2). \tag{21}$$

Since the contrast parameter γ is assumed to be 1, we will have

$$G_0^{(i)} = 1, \quad \text{and} \quad G_1^{(i)} = G_2^{(i)} = G_3^{(i)} = 0. \tag{22}$$

The coefficients $H_1^{(i)}, H_2^{(i)}, H_3^{(i)}, H_4^{(i)}$ for the brightness parameters are given in APPENDIX F. The coefficients of the MSE expression (19) for this case can be obtained by substituting $H_1^{(i)}, H_2^{(i)}, H_3^{(i)}, H_4^{(i)}$ and (22) in APPENDIX E.

On the other hand, if frame-to-frame intensity variations can be modeled using the contrast parameter only, the MSE expression in (14) simplifies to

$$\text{MSE}^{(i)} = \frac{1}{N_1 N_2} \sum_{n_1, n_2 \in \mathcal{B}} [\gamma s_1(n_1, n_2) - \tilde{s}_2(n_1 + d_1, n_2 + d_2)]^2, \quad (d_1, d_2) \in \mathcal{Q}^{(i)}. \tag{23}$$

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

$$\gamma^* = \frac{\sum \tilde{s}_2(n_1 + d_1, n_2 + d_2) s_1(n_1, n_2)}{\sum s_1^2(n_1, n_2)}. \tag{24}$$

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

$$H_0^{(i)} = H_1^{(i)} = H_2^{(i)} = H_3^{(i)} = 0. \tag{25}$$

The coefficients $G_1^{(i)}, G_2^{(i)}, G_3^{(i)}, G_4^{(i)}$ for the contrast parameters are given in APPENDIX G. The coefficients of the MSE expression (19) for this case can be obtained by substituting $G_1^{(i)}, G_2^{(i)}, G_3^{(i)}, G_4^{(i)}$ and (25) in Appendix APPENDIX E.

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 B over a specified block of pixels.

2. Compute the coefficients for contrast and brightness parameters, $G_0^{(i)}, \ldots, H_3^{(i)}$, given in APPENDIX D (or APPENDIX F or APPENDIX G depending on the intensity variation model) for each quadrant.

3. Compute the MSE coefficients $C_0^{(i)}, \cdots, C_8^{(i)}$, given in APPENDIX E for each quadrant, i.e., for each $i = 1, 2, 3, 4$.

4. Compute the coefficients $E_0, \cdots, E_5$, of the fifth order polynomial in APPENDIX C for each quadrant.

5. Find the zeroes of (12) 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 (19), using the efficient exhaustive search method and the new definition for the coefficients $C_0^{(i)}, \cdots, C_8^{(i)}$ proposed in this section.

5. FINDING THE AFFINE MOTION PARAMETERS BETWEEN TWO FRAMES

An affine motion includes a wide range of transformations which consists of translation, rotation, shear and scale operations. If there is an affine motion between two frames, we need to estimate six parameters to characterize the motion fully. These six parameters may be estimated using the correspondence of three non-collinear points in both images.[8] If the point (u_i, v_i) in the transformed image corresponds to the point (x_i, y_i) in the reference image for $i = 1, 2, 3$, then the relation between them can be written using a single matrix equation as,

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} = \begin{bmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ u_3 & v_3 & 1 \end{bmatrix} \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix} . \tag{26}$$

Let us name the above matrices as X, U, and T, successively. Then, in order to find the parameters of the affine motion, namely $t_{11}, t_{12}, t_{21}, t_{22}, t_{31}$, and t_{32}, we need to solve $X = UT$ for T.

The proposed subpixel displacement estimation algorithm can be used to find the corresponding point locations, i.e., the coordinates of the points (x_i, y_i) given (u_i, v_i). First, three different blocks of pixels are selected on the transformed image, and the centers of these blocks are chosen as the points (u_1, v_1),(u_2, v_2), (u_3, v_3). Then, the proposed near-closed-form solution is used to estimate the translational motion between the same three blocks of the transformed and the reference images. In order to find the point (x_i, y_i) in the reference image that corresponds to the point (u_i, v_i) in the transformed image, we simply add to (u_i, v_i) the estimated displacement vector found for that block.

As the motion between corresponding blocks of the reference and transformed images are not purely translational, perfect correspondence between the points (x_i, y_i) and (u_i, v_i) may not be established by estimating the translational motion. As a consequence, the affine transformation matrix T obtained using the translation motion vectors may not be correct. The remedy for this problem is to apply translational motion estimation and affine parameter estimation over the same blocks iteratively, until the affine motion parameters between two consecutive iterations are sufficiently close to that of an identity transformation. In each iteration, the found T matrix is cascaded with the previously found matrices and the overall transformation matrix is applied to the original unsteady frame to prevent any accumulation of interpolation errors.

6. EXPERIMENTAL RESULTS

We have tested the proposed near-closed-form method on three image sequences. One of these sequences is generated from a real CT image (shown in Fig. 1(a)). The other two sequences, namely the Text-1 and Text-2 Sequences (Fig. 2), are generated from a synthetic image (shown in Fig. 1(b)). The subsequent 19 frames of all three sequences are generated by introducing random subpixel displacements to the first frames, i.e., the reference frames, shown in Figs. 1(a) and (b).

The CT Sequence does not contain any intensity variations. On the other hand, we have simulated both contrast and brightness variations on the Text-1 Sequence, and only contrast variations on the Text-2 Sequence. In order to introduce the intensity variations to the subsequent frames of Text-1 sequence, the pixel intensities in the frames are first multiplied by γ, and η is added afterwards as explained in (13). If we let k denote the frame number, the values of γ and η chosen for the k^{th} frame are defined in our experiments as

$$\gamma(k) = 1 + \gamma_0 k \tag{27}$$

$$\eta(k) = \eta_0 k \tag{28}$$

For the Text-1 Sequence, we have set $\gamma(k) = 1 - 0.02(k-1)$ and $\eta(k) = -2(k-1)$, whereas for the Text-2 Sequence, we have set $\gamma(k) = 1 - 0.04(k-1)$ and $\eta(k) = 0$. In addition to intensity variations, a 10 dB white Gaussian noise is added to each frame in the Text-1 and Text-2 Sequences to simulate any observation noise. A 5 × 5 uniform blur is applied to each frame prior to subpixel displacement estimation to reduce the effects of bilinear interpolation and of any additive noise. The simulations are carried on a block of size approximately 100 × 100 pixels, that contains sufficient intensity variations.

In Fig. 3, we compare the performance of the proposed near-closed-form solution to that of the spatio-temporal differentiation method, phase and cross correlation surface interpolation methods and the exhaustive search method (the accuracy of the exhaustive search method is chosen to be $\frac{1}{16}$ pixels). We observe that the cross correlation method performs nearly as well as the proposed near-closed-form solution for the Text-1 and Text-2 Sequences. However, the performance of the cross correlation method degrades significantly for the CT Sequence. On the other hand, the performance of the differentiation method is close to that of the proposed near-closed-form solution for the CT and the Text-1 Sequences, while it degrades considerably for the Text-2 Sequence. We conclude from Fig. 3 that the proposed near-closed-form solution consistently gives the best results for all the sequences considered.

In Fig. 4, the plots of $\gamma(k)$ and $\eta(k)$ versus the frame number (k) are given for the Text-1 Sequence. It can be seen that the near-closed-form solution is able to estimate the parameters $\gamma(k)$ and $\eta(k)$ for each frame (k) almost perfectly.

The CPU times for all of the methods compared in Fig. 3 are nearly the same, except for the exhaustive search method, which is significantly slower than the others.

We have also carried out experiments to test the performance of the proposed affine motion estimation algorithm on the CT image shown in Figure 1(a). The transformed image is the counter-clockwise rotated version of this reference image by 10 degrees. The sizes of the three blocks are chosen as 79 × 79. The correct affine motion parameters between the rotated and reference images are given by

$$T^* = \begin{bmatrix} 0.9848 & -0.1736 & 0 \\ 0.1736 & 0.9848 & 0 \\ -40.4787 & 48.2391 & 1 \end{bmatrix}. \tag{29}$$

The affine parameter estimation algorithm is executed five times and the overall affine motion matrix T_k obtained after iteration k is:

$$T_1 = \begin{bmatrix} 0.9790 & -0.1751 & 0 \\ 0.1926 & 1.0124 & 0 \\ -39.2509 & 43.6864 & 1 \end{bmatrix}, T_2 = \begin{bmatrix} 0.9852 & -0.1704 & 0 \\ 0.1779 & 0.9825 & 0 \\ -40.3991 & 47.3248 & 1 \end{bmatrix}, T_3 = \begin{bmatrix} 0.9858 & -0.1738 & 0 \\ 0.1730 & 0.9836 & 0 \\ -39.699 & 48.3858 & 1 \end{bmatrix}, \tag{30}$$

$$T_4 = \begin{bmatrix} 0.9849 & -0.1738 & 0 \\ 0.1732 & 0.9849 & 0 \\ -39.4366 & 48.1345 & 1 \end{bmatrix}, T_5 = \begin{bmatrix} 0.98497 & -0.1737 & 0 \\ 0.1734 & 0.9850 & 0 \\ -39.4734 & 48.0752 & 1 \end{bmatrix}.$$

It can be seen from the above matrices that, as the iteration number k increases, the matrix T_k becomes a closer estimate of the correct matrix T^*. The incremental matrices A_k's after the k^{th} iteration are:

$$A_2 = \begin{bmatrix} 1.0036 & -0.0004 & 0 \\ -0.0152 & 0.9706 & 0 \\ -0.3415 & 4.9069 & 1 \end{bmatrix}, A_3 = \begin{bmatrix} 0.9997 & -0.0032 & 0 \\ -0.0049 & 1.0017 & 0 \\ 0.9193 & 0.8515 & 1 \end{bmatrix}, \tag{31}$$

(a)

(b)

Figure 1. (a) The first frame of the CT Sequence. (b) The first frame of the Text-1 and Text-2 Sequences.

(a)

(b)

Figure 2. (a) The last frame of the Text-1 Sequence. (b) The last frame of the Text-2 Sequence.

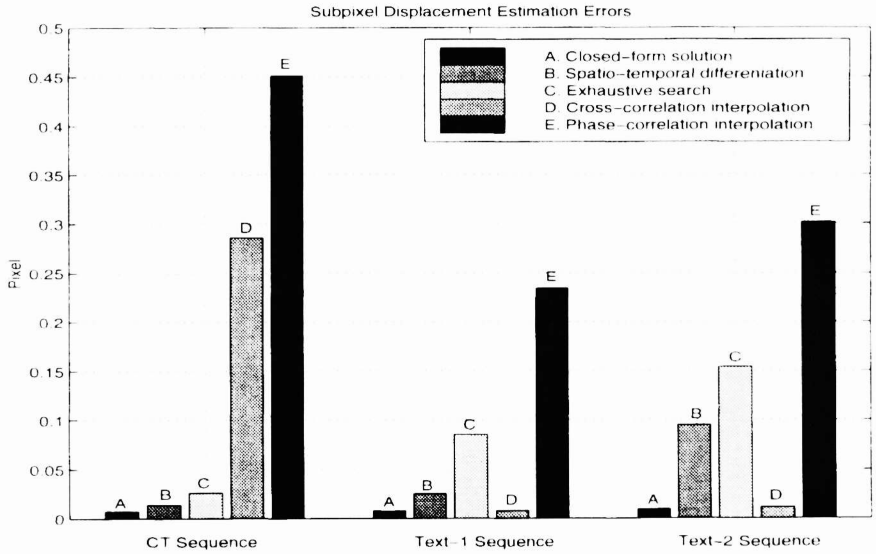

Figure 3. The magnitude of the displacement estimation error vector $(\epsilon(d_1), \epsilon(d_2))$.

$$A_4 = \begin{bmatrix} 0.9992 & 0.0002 & 0 \\ 0.0003 & 1.0013 & 0 \\ 0.2169 & -0.3050 & 1 \end{bmatrix}, A_5 = \begin{bmatrix} 1.0001 & 0.0001 & 0 \\ 0.0002 & 1.0001 & 0 \\ -0.0449 & -0.0568 & 1 \end{bmatrix}.$$

Note that $T_k = A_k A_{k-1} \ldots A_2 T_1$, $k \geq 2$. It can be observed that the off-diagonal parameters of the incremental matrices A_k monotonically get smaller after each iteration and hence A_k approaches to the identity matrix. This shows that, the proposed algorithm converges to the correct affine motion parameters.

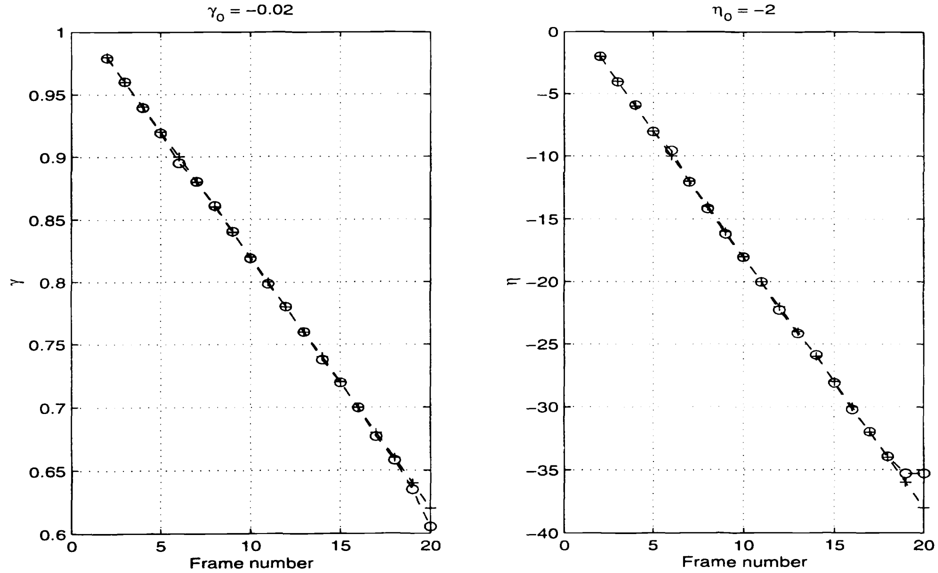

Figure 4. The γ and η values found for the Text-1 Sequence in which $(\gamma_0, \eta_0) = (-0.02, -2)$. The "o" and "+" signs denote the estimated and true values for γ (on the left) and η (on the right), respectively.

7. CONCLUSION

In this paper, we introduced a novel near-closed-form solution for subpixel accuracy stabilization of unsteady image sequences. In the proposed method, the mean-squared error over subpixel displaced frames is minimized with respect to the motion vector components analytically. The method is made robust to intensity variations between frames by modeling such variations using contrast and brightness parameters. If a closed-form solution can not be found, which happened rarely in our experiments, an efficient exhaustive search method is used. The performance of the proposed near-closed-form solution is compared with the spatio-temporal differentiation method, phase and cross correlation surface interpolation methods and the exhaustive search method. It is shown that the near-closed-form solution outperforms the other methods in terms of motion vector estimation errors. Finally, the proposed near-closed-form solution is utilized for estimation of affine motion parameters between two frames, using three point correspondences.

APPENDIX A. 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,

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

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

$$\begin{aligned}
A_{0,0} &= \sum s_1(n_1, n_2), \\
A_{0,0;\ 0,0} &= \sum s_1^2(n_1, n_2), \\
B_{0,0;\ i,j} &= \sum s_1(n_1, n_2) s_2(n_1 + i, n_2 + j), \quad i, j = -1, 0, 1, \\
D_{i,j} &= \sum s_2(n_1 + i, n_2 + j), \quad i, j = -1, 0, 1, \\
D_{i,j;\ k,\ell} &= \sum s_2(n_1 + i, n_2 + j) s_2(n_1 + k, n_2 + \ell), \quad i, j, k, \ell = -1, 0, 1, \quad (i,k), (j,\ell) \neq (\pm 1, \mp 1).
\end{aligned}$$

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

APPENDIX B. MSE COEFFICIENTS

In the following, we express $C_0^{(i)}, \ldots, C_8^{(i)}$, in terms of the basic summations. The scalars I and J are determined by the quadrant number as defined in Equation (6).

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

APPENDIX C. THE COEFFICIENTS OF THE FIFTH ORDER POLYNOMIAL

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

$$
\begin{aligned}
E_0 &= -4C_2C_4^2 + 2C_1C_3C_4 - C_1^2C_6 \\
E_1 &= -8C_4^2C_5 + 4C_1C_4C_7 + 2C_3^2C_4 - 8C_2C_4C_6 - 2C_1^2C_8 \\
E_2 &= 2C_1C_6C_7 - 4C_2C_6^2 + C_3^2C_6 - 8C_2C_4C_8 - 16C_4C_5C_6 - 2C_1C_3C_8 + 6C_3C_4C_7 \\
E_3 &= -8C_2C_6C_8 + 4C_4C_7^2 - 16C_4C_5C_8 + 4C_3C_6C_7 - 8C_5C_6^2 \\
E_4 &= 3C_6C_7^2 - 4C_2C_8^2 - 16C_5C_6C_8 + 2C_3C_7C_8 \\
E_5 &= 2C_7^2C_8 - 8C_5C_8^2
\end{aligned}
$$

APPENDIX D. THE COEFFICIENTS FOR CONTRAST AND BRIGHTNESS PARAMETERS

Let

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

Then

$$
\begin{aligned}
G_0^{(i)} &= \Delta[B_{0,0;\ 0,0} - A_{0,0}D_{0,0}] \\
G_1^{(i)} &= I\Delta[B_{0,0;\ I,0} - B_{0,0;\ 0,0} - A_{0,0}(D_{I,0} - D_{0,0})] \\
G_2^{(i)} &= J\Delta[B_{0,0;\ 0,J} - B_{0,0;\ 0,0} - A_{0,0}(D_{0,J} - D_{0,0})] \\
G_3^{(i)} &= IJ\Delta[B_{0,0;\ 0,0} - B_{0,0;\ I,0} - B_{0,0;\ 0,J} + B_{0,0;\ I,J} - A_{0,0}(D_{0,0} - D_{I,0} - D_{0,J} + D_{I,J})]
\end{aligned}
$$

and

$$
\begin{aligned}
H_0^{(i)} &= \Delta[A_{0,0;\ 0,0}D_{0,0} - A_{0,0}B_{0,0;\ 0,0}] \\
H_1^{(i)} &= I\Delta[A_{0,0;\ 0,0}(D_{I,0} - D_{0,0}) - A_{0,0}(B_{0,0;\ I,0} - B_{0,0;\ 0,0})]
\end{aligned}
$$

$$
\begin{aligned}
H_2^{(i)} &= J\Delta[A_{0,0;\ 0,0}(D_{0,J} - D_{0,0}) - A_{0,0}(B_{0,0;\ 0,J} - B_{0,0;\ 0,0})] \\
H_3^{(i)} &= IJ\Delta[A_{0,0;\ 0,0}(D_{0,0} - D_{I,0} - D_{0,J} + D_{I,J}) \\
&\qquad -A_{0,0}(B_{0,0;\ 0,0} - B_{0,0;\ I,0} - B_{0,0;\ 0,J} + B_{0,0;\ I,J})]
\end{aligned}
$$

In the above, the scalars I and J are again determined by the quadrant number i as given in Equation (6).

APPENDIX E. MSE COEFFICIENTS IN THE CASE OF INTENSITY VARIATIONS

In the following, we give the expressions for $C_0^{(i)}, \ldots, C_8^{(i)}$, in terms of the basic summations. The scalars I and J are determined by the quadrant number as defined in Equation (6).

$$
\begin{aligned}
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} \\
&\qquad +(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} \\
&\qquad +(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} \\
&\qquad +(G_1 + G_2 - G_3 - G_0)B_{0,0;\ 0,0} + (G_0 - G_2)B_{0,0;\ I,0} \\
&\qquad +(G_0 - G_1)B_{0,0;\ 0,J} - G_0B_{0,0;\ I,J} \\
&\qquad +(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} \\
&\qquad +(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} \\
&\qquad +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} \\
&\qquad +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} \\
&\qquad -(G_1 - G_3)B_{0,0;\ 0,0} + (G_1 - G_3)B_{0,0;\ I,0} \\
&\qquad +G_1B_{0,0;\ 0,J} - G_1B_{0,0;\ I,J} - (H_1 - H_3)D_{0,0} \\
&\qquad -D_{0,0;\ 0,0} + 2D_{0,0;\ I,0} + D_{0,0;\ 0,J} \\
&\qquad -D_{0,0;\ I,J} + (H_1 - H_3)D_{I,0} - D_{I,0;\ I,0} \\
&\qquad -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} \\
&\qquad -(G_2 - G_3)B_{0,0;\ 0,0} + (G_2 - G_3)B_{0,0;\ 0,J} \\
&\qquad +G_2B_{0,0;\ I,0} - G_2B_{0,0;\ I,J} - (H_2 - H_3)D_{0,0} \\
&\qquad -D_{0,0;\ 0,0} + 2D_{0,0;\ 0,J} + D_{0,0;\ I,0} \\
&\qquad -D_{0,0;\ I,J} + (H_2 - H_3)D_{0,J} - D_{0,J;\ 0,J} \\
&\qquad -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} \\
&\qquad +2G_3B_{0,0;\ I,0} + 2G_3B_{0,0;\ 0,J} - 2G_3B_{0,0;\ I,J} \\
&\qquad -2H_3D_{0,0} + D_{0,0;\ 0,0} - 2D_{0,0;\ I,0} \\
&\qquad -2D_{0,0;\ 0,J} + 2D_{0,0;\ I,J} + 2H_3D_{I,0} + D_{I,0;\ I,0} \\
&\qquad +2D_{I,0;\ 0,J} - 2D_{I,0;\ I,J} \\
&\qquad +2H_3D_{0,J} - 2H_3D_{I,J} + D_{0,J;\ 0,J} - 2D_{0,J;\ I,J} + D_{I,J;\ I,J}
\end{aligned}
$$

APPENDIX F. THE COEFFICIENTS FOR THE BRIGHTNESS PARAMETER

$$
\begin{aligned}
H_0^{(i)} &= D_{0,0} - A_{0,0} \\
H_1^{(i)} &= I(D_{I,0} - D_{0,0}) \\
H_2^{(i)} &= J(D_{0,J} - D_{0,0}) \\
H_3^{(i)} &= IJ(D_{0,0} - D_{I,0} - D_{0,J} + D_{I,J})
\end{aligned}
$$

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

APPENDIX G. THE COEFFICIENTS FOR THE CONTRAST PARAMETER

$$
\begin{aligned}
G_0^{(i)} &= B_{0,0;\ 0,0}/A_{0,0;\ 0,0} \\
G_1^{(i)} &= I(B_{0,0;\ I,0} - B_{0,0;\ 0,0})/A_{0,0;\ 0,0} \\
G_2^{(i)} &= J(B_{0,0;\ 0,J} - B_{0,0;\ 0,0})/A_{0,0;\ 0,0} \\
G_3^{(i)} &= IJ(B_{0,0;\ 0,0} - B_{0,0;\ I,0} - B_{0,0;\ 0,J} + B_{0,0;\ I,J})/A_{0,0;\ 0,0}
\end{aligned}
$$

In the above, the scalars I and J are again determined by the quadrant number i as given in equation (6).

REFERENCES

1. L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys* **24**, pp. 325–376, 1992.
2. C. D. Kuglin and D. C. Hines, "The phase correlation image alignment method," *Proc. IEEE Int. Conf. Cybernetics Society* , pp. 163–165, 1975.
3. Q. Tian and M. N. Huhns, "Algorithms for subpixel registration," *Computer Vision, Graphics, and Image Processing* **35**, pp. 220–233, 1986.
4. A. M. Tekalp, *Digital Video Processing*, Prentice-Hall, New Jersey, 1995.
5. S. D. Conte and C. D. Boor, *Elementary Numerical Analysis—An Algorithmic Approach*, McGraw-Hill, New York, 1980.
6. Ç. Eroğlu, "Subpixel accuracy image registration with application to unsteadiness correction," Master's thesis, Bilkent University, 1997.
7. C. S. Fuh and P. Maragos, "Affine models for image matching and motion detection," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.* , pp. 2409–2412, 1991.
8. G. Woolberg, *Digital Image Warping*, IEEE Computer Society Press, 1990.