Third order gradient methods for motion estimation

Y. Keller[2], A. Averbuch[1]

[1]School of Computer Science,

Tel Aviv University, Tel Aviv 69978, Israel

[2]Math department

Yale University, Connecticut, USA

Phone: +1-203-432-4345

Email: yosi.keller@yale.edu

Submitted to the IEEE Transactions on Image Processing

October 2002

Abstract

Gradient based motion estimation techniques (GM) are considered to be in the heart of state-of-the-art registration algorithms, being able to account for both pixel and subpixel registration and to handle various motion models (translation, rotation, affine, projective). These methods estimate the motion between two images using a first order Taylor expansion of the registered images. This paper offers two contributions: First, we improve the convergence properties of the GM by an order of magnitude by using a second order Taylor expansion. Second, we extend the third order algorithm using a symmetrical formulation which further improves the convergence properties. The results are verified by theoretical analysis and experimental trials.

Keywords: Global motion estimation, Sub-pixel registration, Gradient methods, image alignment

EDICS Category={2-ANAL, 2-MOTD}

1 Introduction

Image registration plays a vital role in many image processing and computer vision applications such as optical flow computation [12], tracking [21, 22, 23], video compression [11, 14], video enhancement [9], parametric and layered motion estimation [24] , virtual reality [1, 4, 10] and 3D reconstruction [15] to name a few. A comprehensive comparative survey by Barron et. al. [2] found the family of gradient-based motion estimation methods (GM), originally proposed by Horn and Schunck [3] and Lucas and Kanade [12], to perform especially well. The purpose of the GM algorithm is to estimate the parameters vector p associated with the *parametric image registration* problem: starting from pure global translation, image plane rotation, 2D affine, and pseudo-projective (8-parameter flow). These models have been used extensively and are directly estimated from image spatio-temporal derivatives using coarse-to-fine estimation. They search for the best parametric geometric transform that minimizes the square of changes between image intensities (SSD) over the whole image. Several formulations of the gradient methods were suggested. The difference between them lies on how the motion parameters are updated [29]: either by incrementing the motion parameters [12] or incrementing the warp matrix [4]. An updated comprehensive description of these methods was given in [16].

Let $I_1(x,y)$ and $I_2(x,y)$ be the images, which have some common overlap. Each pixel in their common support satisfies

$$I_1(x,y) = I_2\left(\widetilde{x}(x,y,p), \widetilde{y}(x,y,p)\right) \tag{1.1}$$

where the structure of the parameters vector p depends on the type of the estimated motion model. Gathering and solving all the equations associated with pixels in the common support, estimates the *global motion* between the images [4]. Thus, a highly over-constrained linear system (each pixel contributes a linear constraint) is formulated yielding a robust estimate. Gathering the equations related to small image patches estimates *local motion* [12]. Equation (1.1) is solved using non-linear iterative optimization techniques such as Gauss-Newton [27] and Levenberg-Marquardt (LM) [4]. Variational approaches [31], interpret gray value changes as temporal derivatives and minimize variational functionals of the motion fields.

A critical implementation issue concerning the GM is the convergence range and rate when large motions are estimated. GM algorithms are unable to estimate rotations bigger than 30° and affine coefficients bigger then 1.2 . While the estimation of translations and rotations might be bootstrapped by Fourier based methods [13, 36], there is no such reliable solution for affine and projective motions. For example, [32, 33, 34, 35] use affine invariant texture region descriptors for feature correspondence to bootstrap wide basis stereo. As the estimated motion becomes larger, the convergence rate decreases and the GM may diverge to a local minima.

We propose to improve the convergence properties of the GM algorithm using a second order Taylor expansion of the registered images. We analytically show that the convergence properties of the new algorithm are superior to those of the regular GM for large and small motions. Convergence orders of 3 and $\frac{3}{2}$ are achieved for small and large motion estimations, compared to 2 and 1 for the regular GM. Further improvement is achieved by combining the second order expansion with a symmetrical formulation [25].

The paper is organized as follows: the introduction to the GM algorithm is given in section 2. The third order GM formulation is given in section 3 while the symmetric extension is given in section 4. The improved convergence is analytically derived in section 5. The analysis results are verified by the experimental results shown in section 6.

2 Gradient methods based motion estimation

GM methodology [16] estimates the motion parameters p by minimizing the intensity discrepancies between the images $I_1(x,y)$ and $I_2(x,y)$

$$p^* = \arg\min_p \left\{ \sum_{(x_1,y_1)\in S} \left(I_1\left(x_i^{(1)}, y_i^{(1)}\right) - I_2\left(f\left(x_i^{(1)}, y_i^{(1)}, p\right), g\left(x_i^{(1)}, y_i^{(1)}, p\right)\right)\right)^2 \right\} \tag{2.1}$$

where

$$\begin{aligned} x_i^{(2)} &= f(x_i^{(1)}, y_i^{(1)}, p) \\ y_i^{(2)} &= g(x_i^{(1)}, y_i^{(1)}, p), \end{aligned} \tag{2.2}$$

S is the set of coordinates of pixels that are common to I_1 and I_2 in I_1's coordinates and p is the estimated parameters vector. Next we follow the formulation of [1, 4] and solve Eq. (2.1) using non-linear iterative optimization techniques such as Gauss-Newton [27] and Levenberg-Marquardt (LM) [4, 27]. The basic GM formulation and its iterative refinement step are described in sections 2.1 and 2.2, respectively. These are embedded in a multi-resolution scheme given in section 2.3.

2.1 Basic GM formulation

The non-linear optimization of Eq. (2.1) is conducted via a linearization procedure, which is based on a pixel-wise first order Taylor expansion of I_1 in terms of I_2 as a function of the parameter vector p:

$$I_1(x_i^{(1)}, y_i^{(1)}) = I_2(x_i^{(2)}, y_i^{(2)}) + \sum_{p_j \in p} \frac{\partial I_2(x_i^{(2)}, y_i^{(2)})}{\partial p_j} p_j \tag{2.3}$$

where $\frac{\partial I_2(x_i^{(2)}, y_i^{(2)})}{\partial p_j}$ is the partial spatial derivative, $I_1(x_i^{(1)}, y_i^{(1)})$ and $I_2(x_i^{(2)}, y_i^{(2)})$ are the ith corresponding pixel in I_1 and I_2, respectively. By gathering the pixel-wise equations we get the system

$$Hp = I_t \tag{2.4}$$

where

$$\begin{aligned} H_{i,j} &= \frac{\partial I_2\left(x_i^{(2)}, y_i^{(2)}\right)}{\partial p_j} \\ &= \frac{\partial I_2\left(x_i^{(2)}, y_i^{(2)}\right)}{\partial x_2} \frac{\partial f\left(x_i^{(1)}, y_i^{(1)}, p\right)}{\partial p_j} + \frac{\partial I_2\left(x_i^{(2)}, y_i^{(2)}\right)}{\partial y_2} \frac{\partial g\left(x_i^{(1)}, y_i^{(1)}, p\right)}{\partial p_j} \end{aligned} \tag{2.5}$$

and

$$\underline{I_{t_i}} = I_1\left(x_i^{(1)}, y_i^{(1)}\right) - I_2\left(x_i^{(2)}, y_i^{(2)}\right). \tag{2.6}$$

Equation (2.4) is solved using least squares

$$p = \left(H^T H\right)^{-1} H^T I_t \tag{2.7}$$

where H^T is the transpose of H. The expressions for $H^T H$ and $H^T I_t$ are derived analytically by direct calculation:

$$\left(H^T H\right)_{k,j} = \sum_i \frac{\partial I_2\left(x_i^{(2)}, y_i^{(2)}\right)}{\partial p_k} \frac{\partial I_2\left(x_i^{(2)}, y_i^{(2)}\right)}{\partial p_j} \tag{2.8}$$

and

$$\left(H^T I_t\right)_k = \sum_i \frac{\partial I_2\left(x_i^{(2)}, y_i^{(2)}\right)}{\partial p_k} I_t^{(i)}. \tag{2.9}$$

2.1.1 Algorithm flow

The basic GM iteration operates as follows:

1. The matrix $H^T H$ and vector $H^T I_t$ are computed using Eq. (2.8) and Eq. (2.9), respectively.
2. Eq. (2.7) is solved and p is returned as the result.

2.2 Iterative solution of gradient methods

Denote:

p_0	an initial estimated solution of Eq. (2.1) given as input, such that $Warp\left(I_2, p_0\right) \approx I_1$
p_n	the estimated solution after n iterations

The nth iteration of the is given by

1. The input image I_2 is warped towards I_1 using the current estimate p_n $n \geq 0$ and it is stored in $\widetilde{I}_2$. p_0 is a given input.
2. I_1 and $\widetilde{I}_2$ are used as input images to the procedure described in section 2.1.
3. Δp, the result of step 2 is used to update the solution

$$p_{n+1} = \Delta p + p_n \qquad n \geq 0.$$

4. Go back to step 1 until one of the following stopping criteria is met
   (a) At most $N_{\max}$ iterations are performed

   or

   (b) The process is stopped if the translation parameters within the updated term Δp reach a predetermined threshold.

2.3 Multiscale gradient methods

In order to improve the convergence properties of the GM, the iterative process is embedded in a coarse-to-fine multiscale formulation [1]. Next we describe the coarse-to-fine formulation.

1. The input images I_1 and I_2 are smoothed. Our experience shows that a separable averaging filter is suitable.

2. I_1 and I_2 are downsampled through a multiscale decomposition, until a minimal size of their mutual area is reached. The size of the minimal common area depends on the motion model.

3. Starting with the coarsest scale, the initial estimate p_0 is used to bootstrap the iterative refinement algorithm described in section 2.2.

4. Iterative refinement (step 3) is applied from coarse-to-fine.

2.4 Robust GM formulations

Robust estimators were introduced to improve the accuracy of the solutions of Eq. (2.4). The median absolute deviation (MAD) estimator was used in [18, 19] to develop an outlier rejection scheme, where equations related to high alignment errors were discarded. Dufaux et. al. [14] proposed a similar scheme for global motion estimation in video compression, where Eq. (2.4) is solved using a truncated quadratic error measure instead of the standard quadratic measure. The method is robust to outliers, which would otherwise bias the estimation. Total least squares algorithms were introduced in [20, 30], they accounts for errors induced by either the finite differences estimate of the image spatial derivatives (H matrix) or the measurements' errors in Eq. (2.4).

3 Third order gradient methods

This section introduces the third order GM formulation (O3GM) which is integrated into the regular GM algorithm presented in section 2. The O3GM replaces *only* the basic GM step given in section 2.1. The iterative refinement (section 2.2) and multiscale steps (section 2.3) are left intact.

The direct relation between the approximation error of the GM and the convergence rate [25, 27] shows that the smaller the approximation error of the minimized cost function, the better the convergence rate. For large motions the approximation error grows, lowering the convergence rate up to a point where the algorithm diverges. Hence, by lowering the approximation error, larger motions can be estimated and small motions need fewer iterations.

In sake of simplicity, we start by presenting 1D translation registration in section 3.1 and extend to general motion and 2D signals in section 3.2.

3.1 1D formulation

We consider the registration of two one-dimensional signals $I_1(x)$ and $I_2(x)$ using the translation motion model

$$I_1\left(x_i^{(1)}\right) = I_2\left(x_i^{(2)}\right) \tag{3.1}$$

where $x_i^{(1)}$ and $x_i^{(2)}$ are the estimates of the coordinate of the ith sample common to I_1 and I_2 satisfying

$$x_i^{(1)} = x_i^{(2)} + \Delta x. \tag{3.2}$$

The translation parameters Δx corresponds to the parameter vector p in the general 2-D case. Similar to section 2, Eq. (3.1) is solved by expanding I_2 using a Taylor series and solving for Δx. We propose to use a second order expansion

$$I_1\left(x_i^{(1)}\right) = I_2\left(x_i^{(2)}\right) + \frac{\partial I_2\left(x_i^{(2)}\right)}{\partial x} \cdot \Delta x + \frac{\partial^2 I_2\left(x_i^{(2)}\right)}{\partial x^2} \cdot \Delta x^2 \tag{3.3}$$

instead of the first order expansion used by the GM algorithm.

3.1.1 Error analysis

Denote by ε_{O3GM} the error associated with Eq. (3.3)

$$\varepsilon_{O3GM}\left(x_i^{(2)}, \Delta x\right) = \frac{1}{6}\frac{\partial^3 I_2\left(\widetilde{x}^{(2)}\right)}{\partial x^3}\Delta x_i^3, \ \ \widetilde{x} \in [0, \Delta x] \tag{3.4}$$

and $\varepsilon_{GM}\left(x_i^{(2)}, \Delta x\right)$ the error associated with 1D Taylor expansion

$$\varepsilon_{GM}\left(x_i^{(2)}, \Delta x\right) = \frac{1}{2}\frac{\partial^2 I_2\left(\widetilde{x}^{(2)}\right)}{\partial x^2}\Delta x_i^2, \ \ \widetilde{x} \in [0, \Delta x]\,. \tag{3.5}$$

Then

$$\left\|\varepsilon_{O3GM}\left(x_i^{(2)}, \Delta x\right)\right\|_{L^p} < \left\|\varepsilon_{GM}\left(x_i^{(2)}, \Delta x\right)\right\|_{L^p}. \tag{3.6}$$

3.1.2 Algorithm flow

Equation (3.3) is valid for any point in the mutual support of I_1 and I_2. Hence, a second order polynomial equation set is derived

$$\left\{\begin{array}{c} a_0^{(1)} + a_1^{(1)} \cdot \Delta x + a_2^{(1)} \cdot \Delta x^2 = 0 \\ \vdots \\ a_0^{(i)} + a_1^{(i)} \cdot \Delta x + a_2^{(i)} \cdot \Delta x^2 = 0 \\ \vdots \\ a_0^{(N)} + a_1^{(N)} \cdot \Delta x + a_2^{(N)} \cdot \Delta x^2 = 0 \end{array}\right. \tag{3.7}$$

where $\left(a_0^{(i)}, a_1^{(i)}, a_2^{(i)}\right)$ are defined according to Eq. (3.3)

$$\begin{aligned} a_0^{(i)} &= I_2\left(x_i^{(2)}\right) - I_1\left(x_i^{(1)}\right) \\ a_1^{(i)} &= \frac{\partial I_2\left(x_i^{(2)}\right)}{\partial x} \\ a_2^{(i)} &= \frac{\partial^2 I_2\left(x_i^{(2)}\right)}{\partial x^2} \end{aligned} \tag{3.8}$$

and N is the size of the common interval. Equation (3.7) is solved by least-squares where

$$f(x) \triangleq \sum_{i=1}^{N} \left[r_i(\Delta x)\right]^2 \tag{3.9}$$

and

$$r_i(\Delta x) \triangleq a_0^{(i)} + a_1^{(i)} \cdot \Delta x + a_2^{(i)} \cdot \Delta x^2. \tag{3.10}$$

Denote J^{1D} and H^{1D} the Jacobian and Hessian matrices, respectively, of $r_i(\Delta x)$ for the $1D$ case

$$\begin{aligned} J_i^{1D} &= a_1^{(i)} + 2a_2^{(i)} \cdot \Delta x \\ H_i^{1D} &= 2a_2^{(i)}. \end{aligned} \tag{3.11}$$

The iterative solution of Eq. (3.7) is given by either the Gauss-Newton [5]

$$\Delta x_{k+1} = \Delta x_k - \left(J^T J\right)^{-1} J^T \underline{C}, \ k = 0, .. \tag{3.12}$$

or Newton's scheme

$$\Delta x_{k+1} = \Delta x_k - \left(\left(J^{1D}\right)^T J^{1D} + \sum_i H_i^{1D} r_i(\Delta x_k)\right)^{-1} \left(J^{1D}\right)^T H, \ k = 0, .. \tag{3.13}$$

An overall illustration of the O3GM is given in Fig. 1.

Δx_0, the initial guess, is given as an input in the first iteration ($k = 0$). We note that by using a zero initial estimate $\Delta x_0 = 0$ we get

$$J_i^{1D} = a_1^{(i)} = \frac{\partial I_2\left(x_i^{(2)}\right)}{\partial x} \tag{3.14}$$

which is identical to the Jacobian matrix term used in the first order expansion in Eq. (2.5). Thus, the regular GM corresponds to the first iteration of the O3GM with a zero initial guess.

3.2 2D formulation

In this section we formulate the third order 2D motion estimation. Image registration is formulated as a non-linear least-squares optimization problem

$$p^* \triangleq \arg\min_p \left\{ \sum_{(x_1,y_1)\in S} \left(I_2\left(x_i^{(2)}, y_i^{(2)}, p\right) - I_1\left(x_i^{(1)}, y_i^{(1)}\right)\right)^2 \right\} \tag{3.15}$$

and the O3GM is introduced by the motion parameters vector p

$$I_1(x_i^{(1)}, y_i^{(1)}) = I_2(x_i^{(2)}, y_i^{(2)}) + \sum_{j=1}^{N_p} p_j \cdot \frac{\partial I_2(x_i^{(2)}, y_i^{(2)})}{\partial p_j} + \frac{1}{2}\left(\sum_{j=1}^{N_p} p_j \cdot \frac{\partial}{\partial p_j}\right)^2 I_2(x_i^{(2)}, y_i^{(2)}) \tag{3.16}$$

where N_p is the number of motion parameters and $\left(\sum_{j=1}^{N_p} p_j \cdot \frac{\partial}{\partial p_j}\right)^2$ is an operator. For the translation motion model $p = (\Delta x, \Delta y)^T$, $N_p = 2$, we get:

$$\left(\sum_{j=1}^{N_p} p_j \cdot \frac{\partial}{\partial p_j}\right)^2 I_2 = \frac{\partial^2 I_2}{\partial (\Delta x)^2}\Delta x^2 + 2\frac{\partial^2 I_2}{\partial \Delta y \Delta y}\Delta y \Delta x + \frac{\partial^2 I_2}{\partial (\Delta y)^2}\Delta y^2. \tag{3.17}$$

For a general motion with N_p parameters Eq. (3.16) contains N_T terms

$$\begin{aligned} N_T(N_p) &\triangleq 1 + 2N_p + C_{N_p}^2 \\ &= 1 + 2N_p + \frac{N_p(N_p-1)}{2} \end{aligned} \tag{3.18}$$

where we have one scalar, $2N_p$ second order polynomials (without the scalar element) and $C_{N_p}^2$ cross-terms. The first and second partial derivatives with respect to the motion parameters

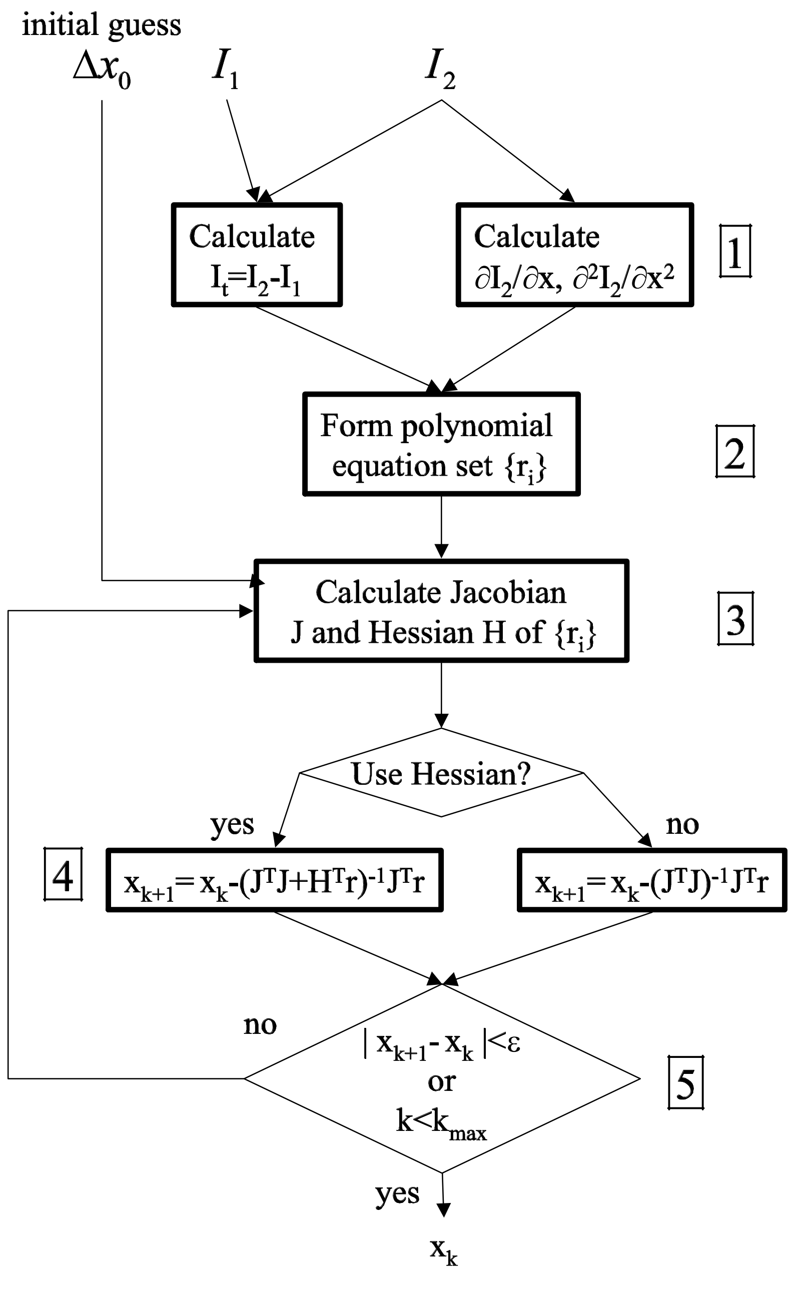

initial guess
Δx_0
I_1
I_2
Calculate $I_t = I_2 - I_1$
Calculate $\partial I_2/\partial x$, $\partial^2 I_2/\partial x^2$
1
Form polynomial equation set $\{r_i\}$
2
Calculate Jacobian J and Hessian H of $\{r_i\}$
3
Use Hessian?
yes
no
4
$x_{k+1} = x_k - (J^T J + H^T r)^{-1} J^T r$
$x_{k+1} = x_k - (J^T J)^{-1} J^T r$
no
$| x_{k+1} - x_k | < \varepsilon$ or $k < k_{max}$
5
yes
x_k

Figure 1: Third order gradient methods. (1) Calculation of the derivatives with respect to the motion parameters. (2) Formation of the polynomial equation set (Eqs. (3.3) and (3.7)). (3) Calculation of the Jacobian and Hessian matrices of the polynomial equation set (Eq. (3.11)). (4) Computation of the iterative refinement using either the Newton (Eq. (3.13)) or Gauss-Newton (Eq. (3.12)). (5) The iterations continue until the stopping criterion is met.

are computed using the chain rule. Equation (3.16) is evaluated for each pixel common to the input images I_1 and I_2 forming the equation set $\{r_i\}_{i=1,\ldots,N}$

$$\begin{aligned} r_i \triangleq\ & a_0^{(i)} \\ & + a_1^{(i)} p_1 + a_2^{(i)} {p_1}^2 + a_3^{(i)} p_2 + a_4^{(i)} {p_2}^2 + \cdots + a_{2N_p-1}^{(i)} p_{N_p} + a_2^{(i)} p_{N_p}^2 \\ & + a_{2N_p+1}^{(i)} p_1 p_2 + \cdots + a_{N_T}^{(i)} p_{N_p} p_{N_{p-1}}. \end{aligned} \tag{3.19}$$

where N is the size of the common area.

Section 8 presents the derivation of the equation set for translations. Similar to section 3.1 Eq. (3.19) is solved using Newtonian methods. We note that the complexity of the computation of the Jacobian and Hessian matrices is negotiable compared to the complexity of the solution of the polynomial equation set.

Denote by J^{2D} and H^{2D} the Jacobian and Hessian matrices, respectively, of $r_i(\Delta x)$ in the $2D$ case. The iterative solution of Eq. (3.19) is given by either the Gauss-Newton formulation

$$p_{k+1} = p_k - \left(\left(J^{2D}\right)^T J^{2D} \right)^{-1} \left(J^{2D}\right)^T r,\ k = 0, .. \tag{3.20}$$

or by Newton's schemes

$$p_{k+1} = p_k - \left(\left(J^{2D}\right)^T J^{2D} + \sum_j H_i^{2D} r_i(p_k) \right)^{-1} \left(J^{2D}\right)^T r,\ k = 0, .. \tag{3.21}$$

where $r = (r_1, \ldots, r_N)^T$, H_i^{2D} is the Hessian matrix and N is the number of corresponding pixels.

3.2.1 Algorithm flow

1. The partial derivatives of I_2 with respect to motion parameters are computed.
2. The second order polynomial equation set $\{r_i\}$ is formed according to Eq. (3.3).
3. J^{2D} and H^{2D} are calculated for Newton's algorithm. But, only J^{2D} is needed for the Gauss-Newton.
4. Either Eq. (3.20) or Eq. (3.21) are evaluated depending on the chosen optimization scheme.
5. Go back to step #4 until one of the following stopping criteria is met:

(a) At most $N_{\max}$ iterations were performed

or

(b) The process is stopped if the translation parameters within the updated term Δp reached a predetermined threshold.

3.2.2 Complexity estimation

The complexity of the O3GM is dominated by the solution of the polynomial equation set by Newton's algorithm. Then each iteration of Newton's algorithm (Eq. (3.20)) consists of

1. Computation of the Jacobian matrix - $N \cdot N_p$ multiplications.
2. Computation of the residual - $N \cdot N_T(N_p) = N \cdot \left(1 + 2N_p + \frac{N_p(N_p-1)}{2}\right)$.
3. Computation of $\left(J^{2D}\right)^T J^{2D}$- $N \cdot 2N_p$.

The over all complexity is $O\left(N \cdot N_p^2\right)$ while the complexity of the GM is $O\left(N \cdot N_p\right)$.

4 Symmetric third order GM formulations

The symmetric GM (SGM) formulation was introduced in [25] to improve the convergence properties of GM algorithm. The symmetry of the image registration problem is used to reduce the approximation error. The error reductions achieved by the O3GM and SGM are complementary, thus, we integrate both in a unified framework we call the Symmetric third order GM (SO3GM).

Similar to section 3.1 we start by considering a 1D registration scenario with a translation motion model. The extension to 2D and general motion models is presented in section 4.2.

4.1 1D formulation

We follow the 1D translation estimation definitions used in section 3.1. In order to lower the estimation error, Eq. (3.1) is reformulated symmetrically with respect to Δx

$$I_1\left(x_i^{(1)} - {}^{\Delta X}\!/_2\right) = I_2\left(x_i^{(2)} + {}^{\Delta X}\!/_2\right). \tag{4.1}$$

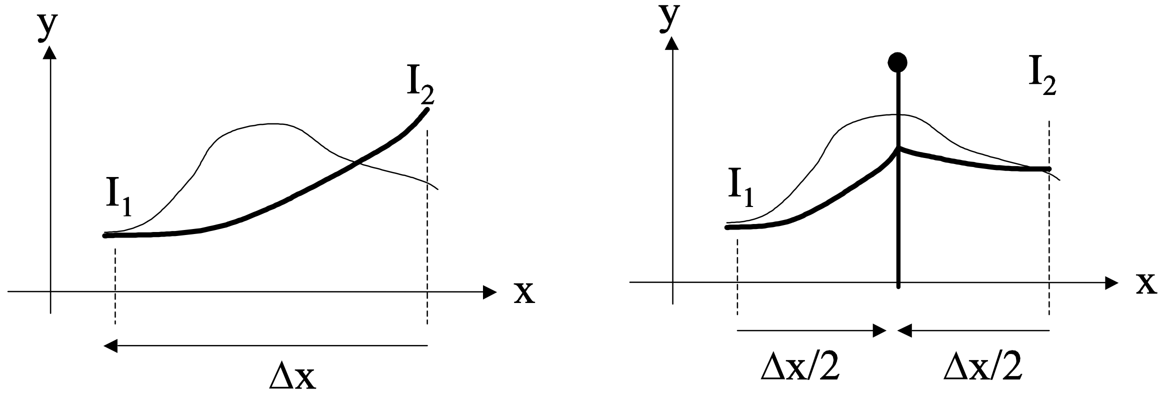


Figure 2: 1D illustration of the approximation used by the Symmetric and non-symmetric Second order gradient methods (O3GM): (a) O3GM: pixels in I_1 are approximated by pixels in I_2 over the interval Δx using a second order approximation of I_2 in the interval. (b) Symmetric O3GM (SO3GM): pixels in the middle of the interval between I_1 and I_2 are approximated by expanding both I_1 and I_2 in a second order approximation.

Following Fig. 2 the r.h.s and l.h.s. of Eq. (4.1) are expanded in a second order Taylor series

$$I_1\left(x_i^{(1)}\right)-\frac{\partial I_1\left(x_i^{(1)}\right)}{\partial x}\cdot\frac{\Delta x}{2}+\frac{\partial^2 I_1\left(x_i^{(1)}\right)}{\partial x^2}\cdot\frac{\Delta x^2}{4}= \\ I_2\left(x_i^{(2)}\right)+\frac{\partial I_2\left(x_i^{(2)}\right)}{\partial x}\cdot\frac{\Delta x}{2}+\frac{\partial^2 I_2\left(x_i^{(2)}\right)}{\partial x^2}\cdot\frac{\Delta x^2}{4} \tag{4.2}$$

$$I_t\left(x_i^{(2)}\right)+\frac{1}{2}\left(\frac{\partial I_2\left(x_i^{(2)}\right)}{\partial x}+\frac{\partial I_1\left(x_i^{(1)}\right)}{\partial x}\right)\cdot\Delta x+\frac{1}{4}\left(\frac{\partial^2 I_2\left(x_i^{(2)}\right)}{\partial x^2}-\frac{\partial^2 I_1\left(x_i^{(1)}\right)}{\partial x^2}\right)\cdot\Delta x^2=0 \tag{4.3}$$

4.1.1 Approximation error analysis

Denote by $\varepsilon_{SO3GM}\left(x_i^{(1)},x_i^{(2)}\right)$ the error associated with Eq. (4.2) then

$$\varepsilon_{SO3GM}\left(x_i^{(1)},x_i^{(2)}\right)\leqslant|\varepsilon_L|+|\varepsilon_R|$$

where ε_L and ε_R are the errors associated with the l.h.s. and r.h.s of Eq. (4.2), respectively. ε_L and ε_R are the approximation errors of a second order Taylor expansion over the interval $[0, {}^{\Delta X}\!/_2]$. Thus we have

$$\begin{aligned}\varepsilon_L = \varepsilon_{O3GM}\left(x_i^{(1)}, {}^{\Delta X}\!/_2\right) = \tfrac{1}{8}\varepsilon_{O3GM}\left(x_i^{(1)}, \Delta x\right)\\ \varepsilon_R = \varepsilon_{O3GM}\left(x_i^{(2)}, {}^{\Delta X}\!/_2\right) = \tfrac{1}{8}\varepsilon_{O3GM}\left(x_i^{(2)}, \Delta x\right)\end{aligned} \tag{4.4}$$

$$\begin{aligned}\varepsilon_{SO3GM}\left(x_i^{(1)}, x_i^{(2)}\right) &\leqslant \frac{1}{8}\varepsilon_{O3GM}\left(x_i^{(1)}, \Delta x\right) + \frac{1}{8}\varepsilon_{O3GM}\left(x_i^{(2)}, \Delta x\right) \\ &\approx \frac{1}{4}\varepsilon_{O3GM}\left(x_i^{(2)}, \Delta x\right)\end{aligned} \tag{4.5}$$

Hence, the approximation error is reduced by a factor of four compared to the O3GM's error. Using the symmetric formulation with the first order GM [25] gives

$$\begin{aligned}\varepsilon_{SGM}\left(x_i^{(1)}, x_i^{(2)}\right) &= \varepsilon_{GM}\left(x_i^{(1)}, {}^{\Delta X}\!/_2\right) \\ &\approx \frac{1}{2}\varepsilon_{GM}\left(x_i^{(2)}, \Delta x\right)\end{aligned} \tag{4.6}$$

4.1.2 Algorithm flow

After constructing Eq. (4.3) for all the corresponding samples between I_1 and I_2, the resulting polynomial equation set is solved using the algorithm described in section 3.1.2.

4.2 2D formulation

In this section we extend the 1D SO3GM presented in the previous section to 2D with general motion models. The image registration problem is formulated symmetrically using a parametric motion model defined by p

$$I_2\left(x_i^{(2)}, y_i^{(2)}, {}^{p}\!/_2\right) = I_1\left(x_i^{(1)}, y_i^{(1)}, -{}^{p}\!/_2\right). \tag{4.7}$$

Both sides of Eq. (4.7) are expanded using a second order Taylor expansion similar to section 3.2

$$\begin{aligned}I_1(x_i^{(1)}, y_i^{(1)}) - \sum_{j=1}^{N_p} \frac{p_j}{2} \cdot \frac{\partial I_1(x_i^{(1)}, y_i^{(1)})}{\partial p_j} + \frac{1}{2}\left(\sum_{j=1}^{N_p} \frac{p_j}{2} \cdot \frac{\partial}{\partial p_j}\right)^2 I_1(x_i^{(1)}, y_i^{(1)}) = \\ I_2(x_i^{(2)}, y_i^{(2)}) + \sum_{j=1}^{N_p} \frac{p_j}{2} \cdot \frac{\partial I_2(x_i^{(2)}, y_i^{(2)})}{\partial p_j} + \frac{1}{2}\left(\sum_{j=1}^{N_p} \frac{p_j}{2} \cdot \frac{\partial}{\partial p_j}\right)^2 I_2(x_i^{(2)}, y_i^{(2)})\end{aligned}$$

and

$$I_t(x_i^{(2)},y_i^{(2)}) + \sum_{j=1}^{N_p} p_j \frac{1}{2}\left(\frac{\partial I_2(x_i^{(2)},y_i^{(2)})}{\partial p_j} + \frac{\partial I_1(x_i^{(1)},y_i^{(1)})}{\partial p_j}\right) + \left(\sum_{j=1}^{N_p} p_j \cdot \frac{1}{8}\frac{\partial}{\partial p_j}\right)^2 \left(I_2(x_i^{(2)},y_i^{(2)}) - I_1(x_i^{(1)},y_i^{(1)})\right) \quad (4.8)$$

where N_p is the number of motion parameters and $\left(\sum_{j=1}^{N_p} p_j \cdot \frac{\partial}{\partial p_j}\right)^2$ is an operator.

4.2.1 Algorithm flow

Equation (4.8) is constructed for all of the pixels in the corresponding support between I_1 and I_2.This polynomial system is solved for p by the procedure given in section 3.2.1.

4.2.2 Complexity estimation

The complexity of the SO3GM is similar to the complexity of the O3GM. Both algorithms solve a polynomial equation set of identical dimension. The complexity of co nstructing the polynomial equation set is negligible compared to its solution.

5 Convergence analysis

This section compares the convergence properties of the GM, O3GM and SO3GM, whose analysis is given in sections 5.1, 5.2 and 5.3, respectively. We show that the proposed O3GM and SO3GM algorithms converge faster than the GM. In sake of simplicity, the analysis of the O3GM and SO3GM is presented using the following 1-D case: Let $g(x) = f(x)^2$ be a non-linear least-squares minimization problem, where

$$x^* = \arg\min_x f(x)^2 .$$

and iteration k of the minimization algorithms is given by

$$x_{k+1} = x_k + \Delta x_k . \quad (5.1)$$

The computation of the iterative refinement term Δx_k differs according to the chosen optimization scheme.

5.1 Convergence properties of the first order GM

A detailed analysis of the first order GM convergence was given in [25], where it was shown that the convergence of the GM can be divided into two steps

Close range: Near the minimum $x_k \longrightarrow x^*, \varepsilon_k \longrightarrow 0$, a quadratic convergence rate is achieved.

Long range: Away from the minimum $\varepsilon_k \gg 0$, a slow linear convergence rate is achieved.

$$\|\varepsilon_{k+1}\| \leq C_{GM}^{LR} \cdot \|\varepsilon_k\| + C_{GM}^{CR} \cdot \|\varepsilon_k\|^2$$

where:

$$\begin{aligned} C_{GM}^{LR} &= \left\| \left[A(x_k) A^T(x_k) \right]^{-1} \sum_{n=1}^{m} \frac{\partial r_n^2(x_k)}{\partial x} r_n(x_k) \right\| \\ C_{GM}^{CR} &= \left\| \left[A(x_k) A^T(x_k) \right]^{-1} \right\| \end{aligned}$$

C_{GM}^{CR}	close range convergence coefficient.
C_{GM}^{LR}	long range convergence coefficient
ε_k	is the parameters estimation error after iteration $k : x^* = x_k + \varepsilon_k$
$r_n(x_k)$	is the error associated with the nth equation at iteration k
$A(x_k)$	is the Jacobian matrix of $f(x)$ at iteration k.

5.2 Convergence properties of the third order GM

In the O3GM scheme, Δx_k is computed by solving the quadratic equation

$$f(x_k) + \frac{\partial f(x_k)}{\partial x} \Delta x_k + \frac{1}{2} \frac{\partial^2 f(x_k)}{\partial x^2} \Delta x_k^2 = 0. \tag{5.2}$$

$f(x^*)$ can be estimated around x_k by a second order Taylor expansion

$$f(x_k) + \frac{\partial f(x_k)}{\partial x} \varepsilon_k + \frac{1}{2} \frac{\partial^2 f(x_k)}{\partial x^2} \varepsilon_k^2 = f(x^*) + \frac{1}{6} \frac{\partial^3 f(\widetilde{x_k})}{\partial x^3}. \tag{5.3}$$

By substituting Eq. (5.2) into Eq. (5.3) we get

$$\frac{\partial f(x_k)}{\partial x} \Delta x_k + \frac{1}{2} \frac{\partial^2 f(x_k)}{\partial x^2} \Delta x_k^2 - \frac{\partial f(x_k)}{\partial x} \varepsilon_k - \frac{1}{2} \frac{\partial^2 f(x_k)}{\partial x^2} \varepsilon_k^2 = -f(x^*) - \frac{1}{6} \frac{\partial^3 f(\widetilde{x_k})}{\partial x^3}. \tag{5.4}$$

Similar to section 5.1, the convergence properties are analyzed for the close and long ranges.

5.2.1 Long range

In the long range $\Delta x_k \gg 0$ and $\varepsilon_k \gg 0$. Hence

$$\frac{1}{2}\left|\frac{\partial^2 f}{\partial x^2}(x_k)\,\Delta x_k^2\right| \gg \left|\frac{\partial f(x_k)}{\partial x}\Delta x_k\right|$$

and

$$\left|\frac{1}{6}\frac{\partial^3 f(\widetilde{x_k})}{\partial x^3}\right| \gg \frac{1}{2}\left|\frac{\partial^2 f(x_k)}{\partial x^2}\Delta x_k^2 + \frac{\partial f(x_k)}{\partial x}\varepsilon_k - f(x^*)\right|.$$

Equation (5.4) is reduced to

$$\left|\frac{1}{6}\frac{\partial^3 f(\widetilde{x_k})}{\partial x^3}\right|\varepsilon_k^3 = \left|\frac{1}{2}\frac{\partial^2 f(x_k)}{\partial x^2}\right|\Delta x_k^2 \tag{5.5}$$

and the iterative refinement term becomes

$$\Delta x_k \approx \sqrt{\left|\frac{1}{3}\left(\frac{\partial^2 f}{\partial x^2}\right)^{-1}\frac{\partial^3 f(\widetilde{x_k})}{\partial x^3}\right|}\varepsilon_k^{3/2}. \tag{5.6}$$

By substituting Eq. (5.6) into Eq. (5.1), we get

$$|\varepsilon_{k+1}| = \sqrt{\left|\left(\frac{\partial^2 f}{\partial x^2}\right)^{-1}\frac{\partial^3 f(\widetilde{x_k})}{\partial x^3}\right|}\,|\varepsilon_k|^{3/2} = C_{O3GM}^{LR}\,|\varepsilon_k|^{3/2}. \tag{5.7}$$

5.2.2 Close range:

Following Eq. (5.4) and using

$$\varepsilon_{k+1} = \varepsilon_k - \Delta x_k$$

we have

$$\frac{\partial f(x_k)}{\partial x}\varepsilon_{k+1} + \frac{1}{2}\frac{\partial^2 f(x_k)}{\partial x^2}(\varepsilon_k + \Delta x_k)\,\varepsilon_{k+1} = f(x^*) + \frac{1}{6}\frac{\partial^3 f(\widetilde{x_k})}{\partial x^3}\varepsilon_k^3. \tag{5.8}$$

Substituting the first order Taylor approximation of $\frac{\partial f(x^*)}{\partial x}$

$$\frac{\partial f(x_k)}{\partial x} + \frac{\partial^2 f(x_k)}{\partial x^2}\varepsilon_k \approx \frac{\partial f(x^*)}{\partial x} = 0$$

into Eq. (5.8), we get

$$\frac{1}{2}\frac{\partial^2 f(x_k)}{\partial x^2}\underbrace{(\varepsilon_k - \Delta x_k)}_{\varepsilon_{k+1}}\varepsilon_{k+1} = f(x^*) + \frac{1}{6}\frac{\partial^3 f(\widetilde{x_k})}{\partial x^3}\varepsilon_k^3. \tag{5.9}$$

Substituting the second order Taylor approximation of $f(x^*)$

$$f(x^*) - \frac{1}{2}\frac{\partial^2 f(x_k)}{\partial x^2}\varepsilon_{k+1}^2 = \frac{\partial f(x_k)}{\partial x}\varepsilon_{k+1}$$

into Eq. (5.9), we have

$$\frac{\partial f(x_k)}{\partial x}\varepsilon_{k+1} = \frac{1}{6}\frac{\partial^3 f(\widetilde{x_k})}{\partial x^3}\varepsilon_k^3.$$

Hence, the O3GM achieves a third order convergence rate in the proximity of the solution

$$|\varepsilon_{k+1}| = C_{O3GM}^{CR}|\varepsilon_k|^3. \tag{5.10}$$

5.3 Convergence properties of the Symmetric third order GM

In section 4.1.1 the SO3GM formulation was shown to achieve an approximation error lower by a factor of four than the O3GM. Hence, the results of the convergence analysis in section 5.2 can be applied by dividing the error term in Eq. (5.3) by four. For the long range we get

$$|\varepsilon_{k+1}| = \sqrt{\left(\frac{\partial^2 f}{\partial x^2}\right)^{-1}\frac{1}{4}\frac{\partial^3 f(\widetilde{x_k})}{\partial x^3}}\,|\varepsilon_k|^{3/2} = \frac{1}{2}C_{O3GM}^{LR}|\varepsilon_k|^{3/2}. \tag{5.11}$$

Thus,

$$C_{SO3GM}^{LR} = \frac{1}{2}C_{O3GM}^{LR} \tag{5.12}$$

and using Eq. (5.10) we get the following for the close range

$$C_{SO3GM}^{CR} = C_{O3GM}^{CR} \tag{5.13}$$

These results, are similar to [25], where it was shown that the symmetric first order GM improves the convergence rate for the long range, but not for the close range.

6 Experimental Results

This section describes the performance of the proposed algorithms and verifies the convergence analysis given in section 5. The same implementations of the *iterative refinement* and *multiscale embedding* [16] were used for the O3GM, SO3GM, SGM and GM algorithms. Thus, the only difference is the *single iteration module*, which was replaced by the SO3GM and SGM algorithms. The translation and rotation simulations were conducted using the *lena* and *airfield* images [25] while the affine and projective motion were tested using real images. The GM algorithm was implemented according to [9, 16] which are considered a state-of-the-art implementation.

The numerical results are expressed in terms of the alignment error vs. the number of iterations needed for convergence. As we showed, the convergence rate (iterations-wise) is related to the robustness of the algorithm. An iterative optimization algorithm which convergence in fewer iterations has a better condition number [27] and will be better able to handle large motions and noisy input.

6.1 Rotation and translation estimation

Figures 3, 4 and 5 show the results of translation and rotation estimation. Registration results for small rotations are presented in Figs. 3 and 4. The SO3GM and O3GM converged significantly faster than the first order formulations (GM, SGM). The SO3GM converged significantly faster than the O3GM: six iterations compared to thirteen, respectively, while being five times faster than the regular GM. The convergence of large motions is presented in Fig. 5 where the SO3GM was significantly superior convergence to the other registration modes.

6.2 Affine and projective motion estimation

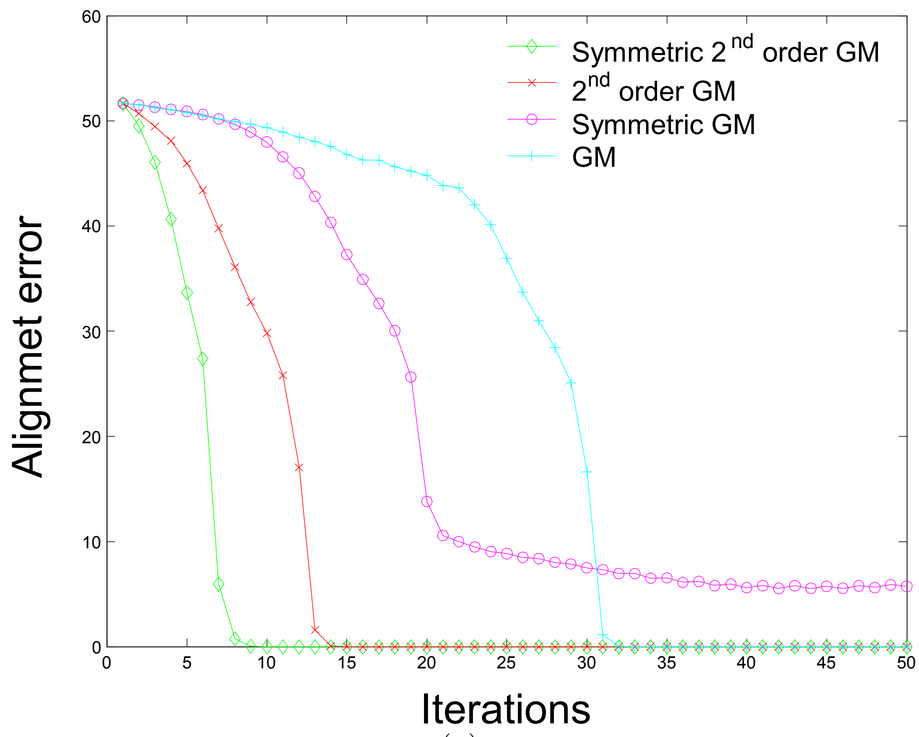
The registration of real affine and projective images are given in Figs. 6 and 7, respectively. The affine motion model is given by

$$\begin{aligned} x_1 &= a \cdot x_2 + b \cdot y_2 + c \\ y_1 &= d \cdot x_2 + e \cdot y_2 + f \end{aligned} \tag{6.1}$$

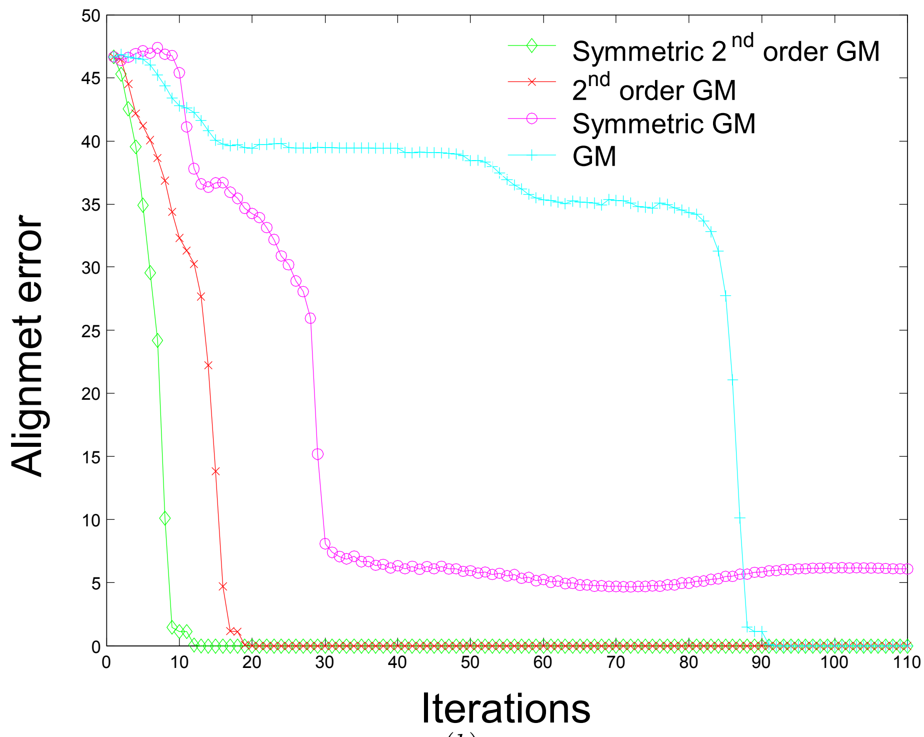
and the projective model is given by:

$$\begin{aligned} x_1 &= \tfrac{a \cdot x_2 + b \cdot y_2 + c}{g \cdot x_2 + h \cdot y_2 + 1} \\ y_1 &= \tfrac{d \cdot x_2 + e \cdot y_2 + f}{g \cdot x_2 + h \cdot y_2 + 1} \end{aligned} . \tag{6.2}$$

In both cases, the O3GM outperformed the GM formulation by converging twice as fast. In the projective case (Fig. 7) the SGM converged slightly better than the O3GM, similar to the

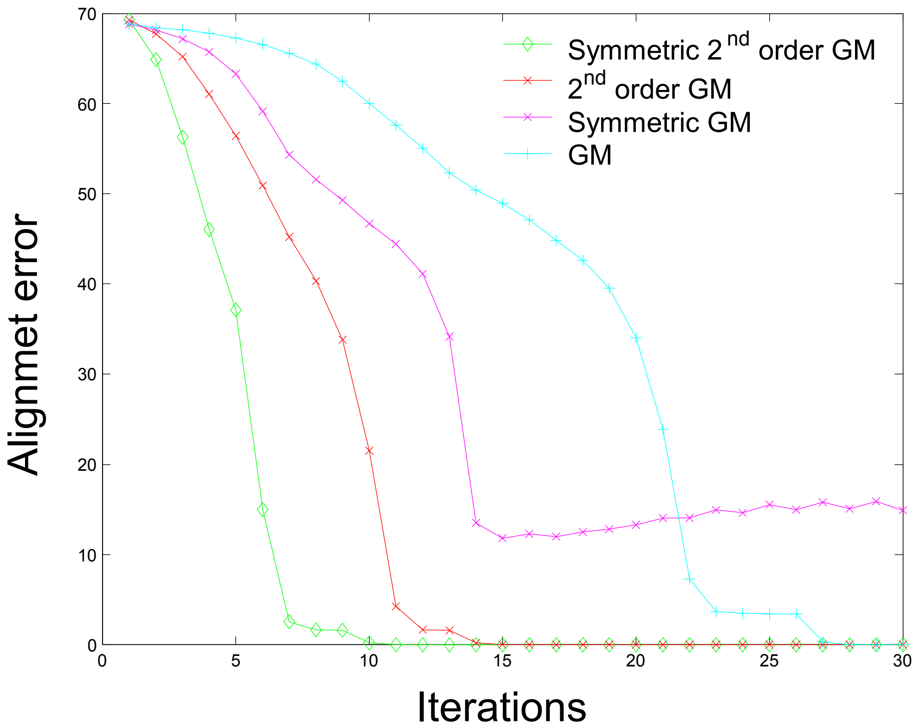


(a)

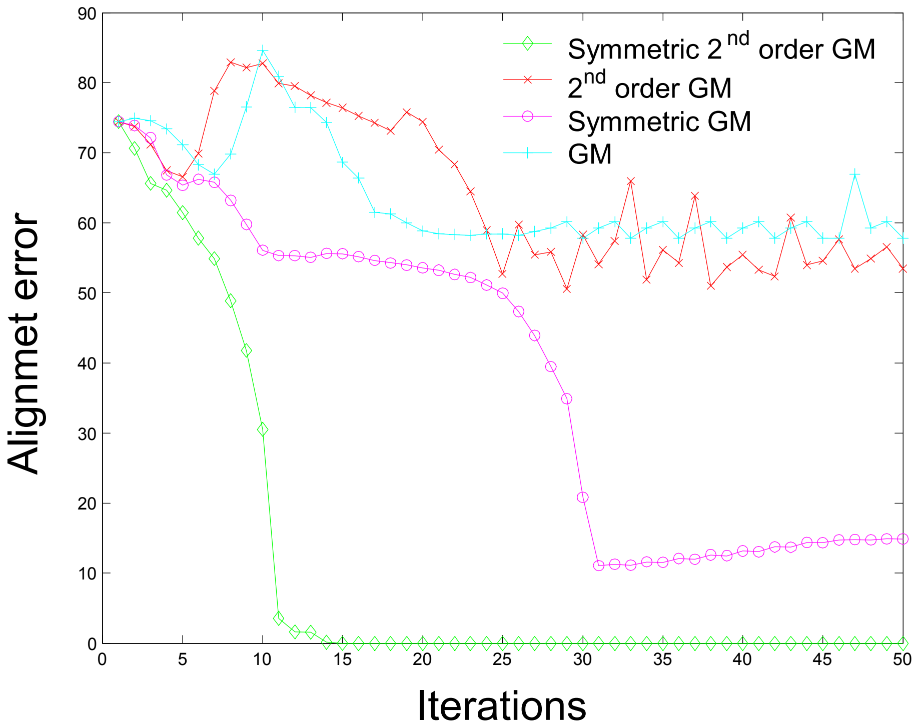


(b)

Figure 3: Rotation and translation estimation results for the *airfield* image. The image was rotated and translated by (a) $(\theta = 5^\circ,\ \Delta x = 20,\ \Delta y = 20)$. (b) $(\theta = 10^\circ,\ \Delta x = 35,\ \Delta y = 35)$. The second order formulations converged significantly faster than the first order formulations. The SO3GM algorithm converged twice as fast as the O3GM algorithm.

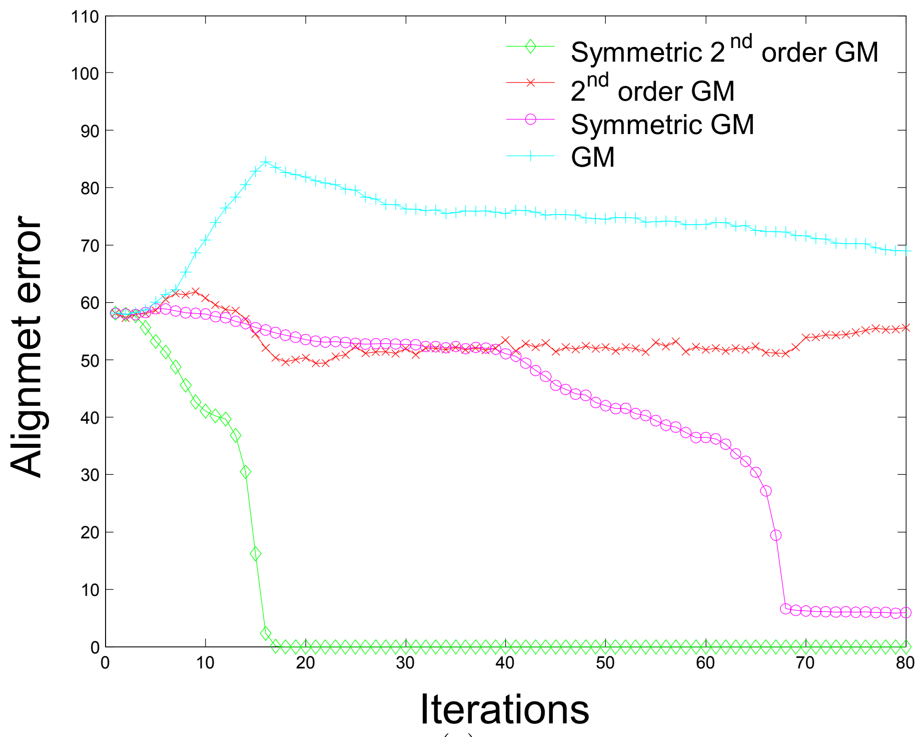


(a)

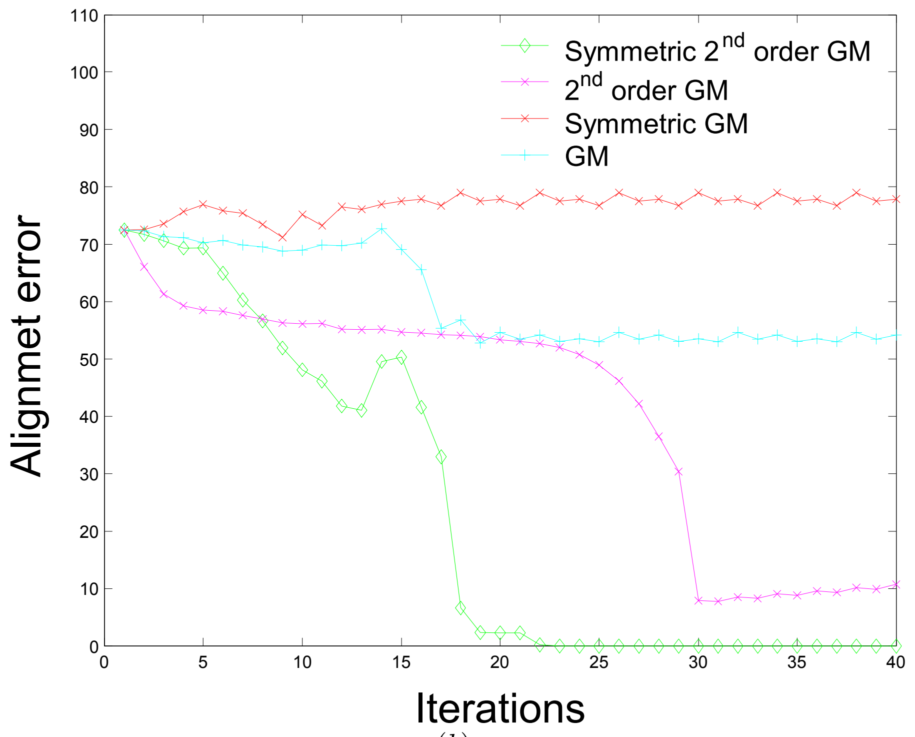


(b)

Figure 4: Registration results for the *lena* image. The image was rotated and translated by (a) $(\theta = 5^\circ,\ \Delta x = 20,\ \Delta y = 20)$. (b) $(\theta = 10^\circ,\ \Delta x = 20,\ \Delta y = 20)$. The SO3GM converged significantly faster than the SGM. The other algorithms diverged.



(a)



(b)

Figure 5: Registration results for large motions. (a) *airfield* image rotated and translated by $(\theta = 15^\circ,\ \Delta x = 0,\ \Delta y = 0)$. (b) *lena* image rotated and translated by $(\theta = 20^\circ,\ \Delta x = 0,\ \Delta y = 0)$. In both cases the SO3GM converged significantly faster than the SGM. The other algorithms diverged.

results of section 6.1, where we attributed the phenomenon to the statistical properties of the texture. We note that for these high-order models the complexity of the proposed algorithms is significantly higher than the complexity of the GM and SGM. The registration of the projective images took 30 seconds on a 1.5 GHz P4 where the algorithm was implemented using non optimized C++.

To conclude, the experimental results verify the convergence analysis - in all cases the O3GM and SO3GM converge faster (iteration-wise) than the regular GM and in most cases, it was able to converge where the regular GM diverged. Due to its high computational complexity it is not recommended for the small motions estimation. It excels in the estimation of large motions where the regular GM might diverge or converge extremely slow.

7 Conclusions and future work

In this paper we proposed two new formulations which enhance the performance of gradient based image registration methods. These algorithms extend the current state-of-the-art image registration algorithms and were proven to possess superior convergence range and rate. Future work includes the application of the O3GM and SO3GM to computer vision problems which are know to be unstable such as wide baseline stereo [35] and 3D reconstruction [15].

8 Appendix A: Third order translation estimation

This section presents the derivation of the O3GM and SO3GM formulations for the 2D translation motion model.

$$I_1(x,y) = I_2(x+\Delta x, y+\Delta y) \tag{8.1}$$

Expanding Eq. (8.1) using a second order Taylor series expansion in 2D, we get

$$\begin{aligned} I_2(x+\Delta x, y+\Delta y) = I_2(x,y) &+ \frac{\partial I_2(x,y)}{\partial x}\Delta x + \frac{\partial I_2(x,y)}{\partial y}\Delta y \\ &+ \frac{1}{2}\Delta x^2 \frac{\partial^2 I_2(x,y)}{\partial x^2} + \frac{1}{2}\Delta y^2 \frac{\partial^2 I_2(x,y)}{\partial y^2} + \Delta x \Delta y \frac{\partial^2 I_2(x,y)}{\partial y \partial x} \end{aligned} \tag{8.2}$$

$$\begin{aligned} I_1(x,y) = I_2(x,y) &+ \frac{\partial I_2(x,y)}{\partial x}\Delta x + \frac{\partial I_2(x,y)}{\partial y}\Delta y \\ &+ \frac{1}{2}\Delta x^2 \frac{\partial^2 I_2(x,y)}{\partial x^2} + \frac{1}{2}\Delta y^2 \frac{\partial^2 I_2(x,y)}{\partial y^2} + \Delta x \Delta y \frac{\partial^2 I_2(x,y)}{\partial y \partial x} \end{aligned} \tag{8.3}$$

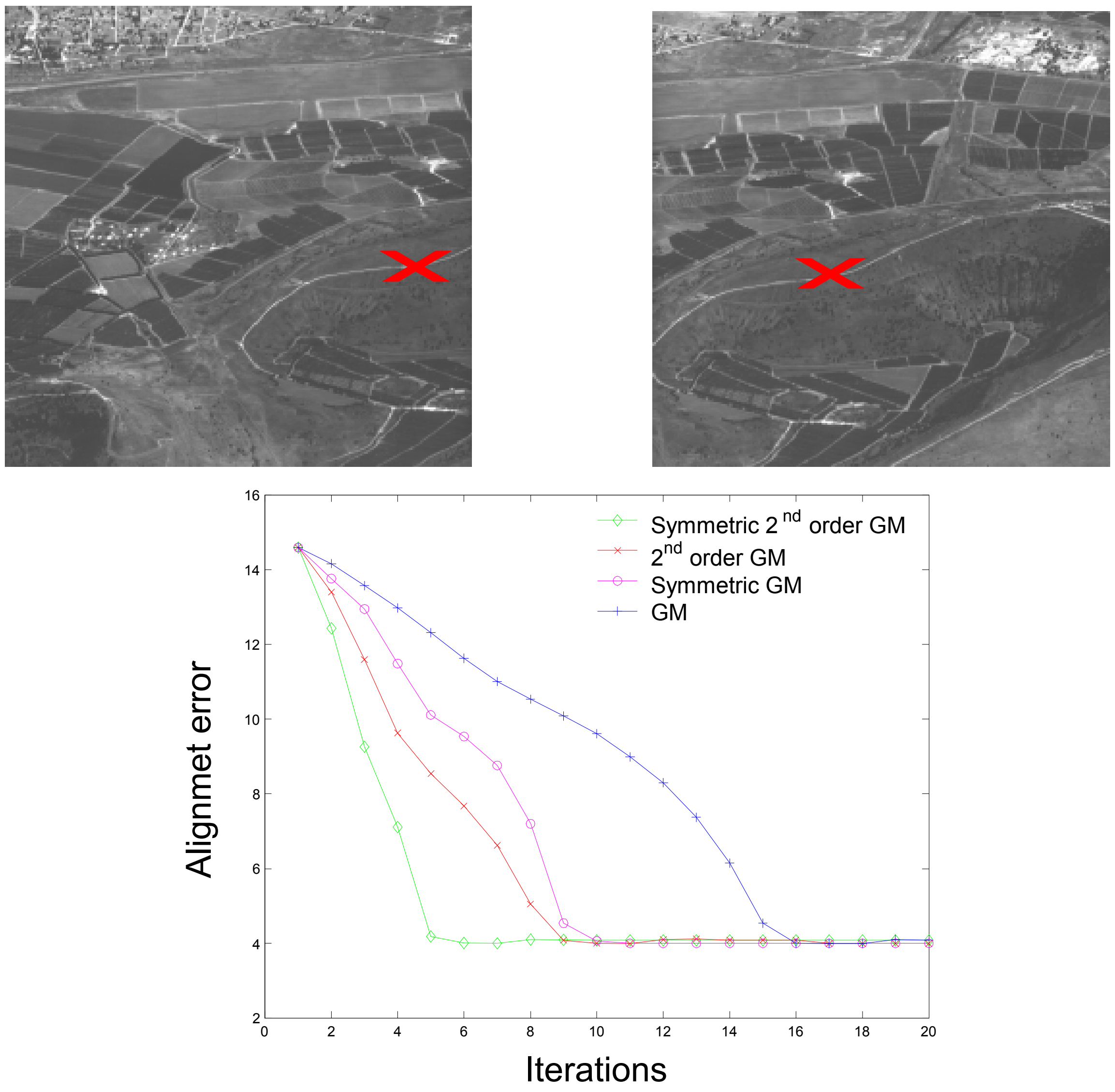


Figure 6: Registration results for affine motion. The initial estimation of the motion was given by computing the relative translation of the pixels marked by the red X. The second order formulations converged significantly faster than the first order algorithms. The SO2GN converged 4 times faster than the GM and twice faster than the O3GM and SGM.

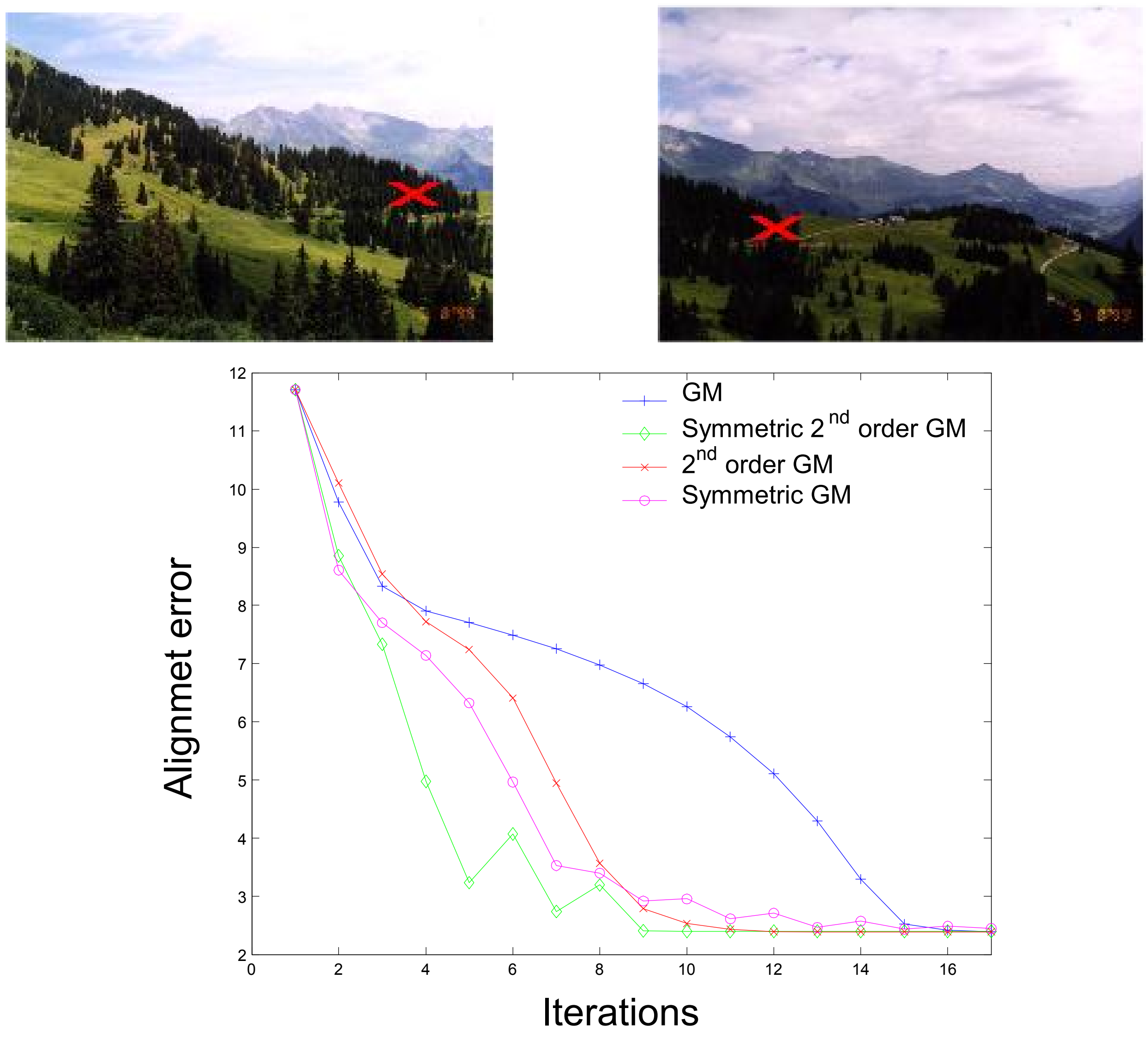


Figure 7: Registration results for projective motion. The initial estimation of the motion was given by computing the relative translation of the pixels marked by the red X. The second order formulations converged significantly faster than the first order formulation. Similar to Fig. 5, the SGM converged better than the O3GM while the SO3GM showed the best convergence properties.

Eq. (8.3) is substituted into Eq. (3.16) in section 3.2 and used for the third order motion estimation.

8.1 Symmetric formulation

In the symmetric formulation we solve

$$I_1\left(x_i^{(1)}-\frac{\Delta x}{2},y_i^{(1)}-\frac{\Delta y}{2}\right)=I_2\left(x_i^{(2)}+\frac{\Delta x}{2},y_i^{(2)}+\frac{\Delta y}{2}\right) \tag{8.4}$$

instead of Eq. (8.1).

By setting $\left(\Delta x=\frac{\Delta x}{2},\Delta y=\frac{\Delta y}{2}\right)$ and substituting into (8.2) we evaluate the r.h.s. of Eq. (8.4)

$$\begin{aligned}I\left(x+\frac{\Delta x}{2},y+\frac{\Delta y}{2}\right)=I\left(x,y\right)+\frac{\partial I\left(x,y\right)}{\partial x}\frac{\Delta x}{2}+\frac{\partial I\left(x,y\right)}{\partial y}\frac{\Delta y}{2}\\+\frac{1}{8}\Delta x^2\frac{\partial^2 I\left(x,y\right)}{\partial x^2}+\frac{1}{8}\Delta y^2\frac{\partial^2 I\left(x,y\right)}{\partial y^2}+\frac{1}{4}\Delta x\Delta y\frac{\partial^2 I\left(x,y\right)}{\partial y\partial x}\end{aligned}$$

The l.h.s. is evaluates similarly by setting $\left(\Delta x=-\frac{\Delta x}{2},\Delta y=-\frac{\Delta y}{2}\right)$ and substituting into (8.3).

Thus we get

$$\begin{aligned}I_1\left(x_1,y_1\right)+\frac{\partial I_1\left(x_1,y_1\right)}{\partial x}\frac{\Delta x}{2}+\frac{\partial I_1\left(x_1,y_1\right)}{\partial y}\frac{\Delta y}{2}+\\\frac{1}{8}\Delta x^2\frac{\partial^2 I_1\left(x_1,y_1\right)}{\partial x^2}+\frac{1}{8}\Delta y^2\frac{\partial^2 I_1\left(x_1,y_1\right)}{\partial y^2}+\frac{1}{2}\Delta x\Delta y\frac{\partial^2 I_1\left(x_1,y_1\right)}{\partial y\partial x}=\\I_2\left(x_2,y_2\right)-\frac{\partial I_2\left(x_2,y_2\right)}{\partial x}\frac{\Delta x}{2}-\frac{\partial I_2\left(x_2,y_2\right)}{\partial y}\frac{\Delta y}{2}+\\\frac{1}{8}\Delta x^2\frac{\partial^2 I_2\left(x_2,y_2\right)}{\partial x^2}+\frac{1}{8}\Delta y^2\frac{\partial^2 I_2\left(x_2,y_2\right)}{\partial y^2}+\frac{1}{4}\Delta x\Delta y\frac{\partial^2 I_2\left(x_2,y_2\right)}{\partial y\partial x}\end{aligned}$$

and the pixelwise equation for the translation model O3GM is given by

$$\begin{aligned}\frac{\Delta x}{2}\left(\frac{\partial I_1\left(x_1,y_1\right)}{\partial x}+\frac{\partial I_2\left(x_2,y_2\right)}{\partial x}\right)+\frac{\Delta y}{2}\left(\frac{\partial I_1\left(x_1,y_1\right)}{\partial y}+\frac{\partial I_2\left(x_2,y_2\right)}{\partial y}\right)\\+\frac{1}{8}\Delta x^2\left(\frac{\partial^2 I_1\left(x_1,y_1\right)}{\partial x^2}-\frac{\partial^2 I_2\left(x_2,y_2\right)}{\partial x^2}\right)+\frac{1}{8}\Delta y^2\left(\frac{\partial^2 I_1\left(x_1,y_1\right)}{\partial y^2}-\frac{\partial^2 I_2\left(x_2,y_2\right)}{\partial y^2}\right)\\+\frac{1}{4}\Delta x\Delta y\left(\frac{\partial^2 I_1\left(x_1,y_1\right)}{\partial y\partial x}-\frac{\partial^2 I_2\left(x_2,y_2\right)}{\partial y\partial x}\right)=I_t\left(x_2,y_2\right)\end{aligned}$$

References

[1] S .Man and R. W. Picard, “Virtual Bellows: constructing high quality stills from video”, Proc. IEEE Int. Conf. Image Processing Austin, TX, pp. 363–367, Nov. 13-16, 1994.

[2] L. Barron, D. J. Fleet and S. S. Beauchemin, “Performance of Optical flow Techniques”, Int. J. Computational Vision, Vol. 12, pp. 43-77, 1994.

[3] B. K. P. Horn and B. G. Schunck, “Determining optical Flow,” Artificial Intelligence,vol. 17, pp. 185-203, 1981.

[4] R. Szeliski, “Image Mosaicking for Tele-Reality Applications”,Workshop on Application of Computer Vision 1994, pp. 44-53, 1994.

[5] P. Gill , “Practical Optimization”, Academic Press, 1982.

[6] A. Averbuch, Y. Keller, “Warp free Gradient methods based image registration”, in preperation.

[7] M. Elad, P. Teo, and Y. Hel-Or, “Optimal Filters For Gradient Based Motion Estimation”, International Conference on Computer Vision - ICCV 1999, Corfu, Greece, pp. 559-565, September 1999.

[8] E. P. Simoncelli, “Design of multi-dimensional derivatives filters,” IEEE International Conf. on Image Processing, Austin Tx, pp. 790-794, 1994.

[9] M. Irani and S. Peleg, “Motion Analysis for Image Enhancement: Resolution, Occlusion and Transparency”. Journal of Visual Communication and Image Representation, Vol. 4, No. 4, pp.324-335, December 1993.

[10] M. Irani, P. Anandan, and S. Hsu. “Mosaic based representations of video sequences and their applications”, International Conference on Computer Vision, pages 605-611, 1995.

[11] A. Tekalp, “Digital Video Processing”, Prentice Hall, 1995.

[12] B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in Proc. DARPA, Image Understanding Workshop, pp. 121-130, 1981.

[13] C. D. Kuglin and D. C. Hines. “The phase correlation image alignment method”, IEEE Conference on Cybernetics and Society, pp. 163–165, September 1975.

[14] F. Dufaux and J. Konrad, ”Efficient, Robust, and Fast Global Motion Estimation for Video Coding”, IEEE Transactions on Image Processing, vol. 9, no. 3, pp. 497-501, March 2000.

[15] G. Stein and A. Shashua. “Model-Based Brightness Constraints: On Direct Estimation of Structure and Motion”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 9, pp. 992-1015, September 2000.

[16] M. Irani and P. Anandan. “All About Direct Methods”, International Workshop on Vision Algorithms, Corfu, Greece, September 1999.

[17] P. J. Burt, R. Hingorani and R. J. Kolezynski, “Mechanism for isolating component patterns in the sequential analysis of multiple motion”, IEEE Workshop on Visual Motion, pp. 187-193, 1991.

[18] A. Fusiello, E. Trucco, T. Tommasini, V. Roberto, “Improving Feature Tracking with Robust Statistics”, Pattern Analysis and Applications Vol. 2, No. 4, 1999, pp. 312-320 .

[19] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto, “Making good features to track better”, in Proceedings IEEE Computer Society Conference on Computer Vision Pattern Recognition, 1998, pp. 145–149.

[20] B. Matei and P. Meer, “Optimal Rigid Motion Estimation and Performance Evaluation with Bootstrap”, Computer Vision and Pattern Recognition Conference (CVPR), Fort Collins, CO, June 1999, vol. 1, pp. 339-345.

[21] Black, M. J. and Jepson, A., “EigenTracking: Robust matching and tracking of articulated objects using a view-based representation”, International Journal of Computer Vision, vol 26 no. 1, pp. 63-84, 1998.

[22] M. L. Cascia, S. Sclaroff and V. Athitsos: “Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models”, IEEE PAMI, Vol. 21, no.6, June 1999.

[23] G.D. Hager and P.N. Bel-humeur. "Efficient region tracking with parametric models of geometry and illumination." IEEE Trans. on Patt. Analysis and Machine Intelligence, Vol. 20 no. 10 pp. 1025–1039, 1998.

[24] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. "Hierarchical model-based motion estimation." In Proc. of ECCV, pages 237–252, 1992.

[25] Y. Keller, A. Averbuch, "Fast motion estimation using bi-directional gradient methods", to appear in the IEEE Trans. On Image Processing. http://www.math.tau.ac.il/~keller/publications/pdf/optical_flow1.pdf

[26] R. Cutler and L. Davis. "Developing Real-Time Computer Vision Applications for Intel Pentium III based Windows NT Workstations," FRAME-RATE: Frame-rate Applications, Methods and Experiences with Regularly Available Technology and Equipment, in conjunction with IEEE International Conference on Computer Vision (ICCV), September 1999, Kerkyra, Greece.

[27] G. Golub and C. Van Loan, "Matrix Computations", 2d Edition, The Johns Hopkins University Press, 1989.

[28] G. Wolberg. "Digital Image Warping". IEEE Computer Society Press, Los Alamitos, California, 1990.

[29] S. Baker and I. Matthews, "Equivalence and Efficiency of Image Alignment Algorithms", Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition, December, 2001.

[30] G. H. Granlund and H. Knutsson. "Signal Processing for Computer Vision". Kluwer Academic, Dordrecht, The Netherlands, 1995.

[31] J. Weickert, C. Schnörr, "Variational Optic Flow Computation with a Spatio-Temporal Smoothness Constraint", Journal Math. Imaging and Vision, Band 14 (2001), Vol. 3, S. 245-255

[32] Vittorio Ferrari, Tinne Tuytelaars, Luc Van Gool "Wide-baseline muliple-view Correspondences", IEEE Computer Vision and Pattern Recognition, Madison, USA, June 2003.

[33] Schaffalitzky, F. and Zisserman, A, “Viewpoint Invariant Texture Matching and Wide Baseline Stereo”, Proc. 8th International Conference on Computer Vision, Vancouver, Canada, month July, 2001.

[34] T. Tuytelaars and L. Van Gool, “Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions.” Proc. BMVC 2000, pages 412–425, Bristol, UK (2000).

[35] J. Matas, O. Chum, M. Urban and Pajdla “Robust Wide Baseline Stereo from Maximally Stable Extremal Regions”, In: Proceedings of the British Machine Vision Conference (BMVC). pp. 384-393. London, GB, 2002.

[36] S. Reddy and B. N. Chatterji. “An FFT-based technique for translation, rotation, and scale-invariant image registration”. IEEE Trans. on Image Processing, 3(8):1266–1270, August 1996.