Fast Registration of Remotely Sensed Images

Arash Abadpour and Shohreh Kasaei
Sharif University of Technology, Tehran, Iran.

***Abstract*— The developments in remote–sensing technologies has offered new opportunities and applications in environmental monitoring. In recent years, researchers have focused on using this information to assess earthquake damages and afterwards reconstructions. A crucial step in change detection processes is to register the multi–spectral temporal images. In this paper, we propose a new, fast, and efficient PCA–based method for remote–sensing image registration. The proposed method works in realtime.**

I. Introduction

In recent years, the spatial and the spectral resolution of the remote–sensing sensors and the revisiting frequency of the satellites, has been extensively increased. These developments, has offered the possibility of addressing new applications of remote–sensing in environmental monitoring. On the other hand, the officials are getting more and more aware of using multi–spectral remote–sensing images for regular and efficient control of the environment [1], [2].

A key issue in remote–sensing image analysis is to detect changes on the earth surface, in order to manage possible interventions to avoid massive environmental problems [3]. Recently, many researchers have worked on using the remote–sensing data to help estimating the earthquake damages [4], [5] or the afterwards reconstruction progresses [6]. Change detection algorithms usually take two sets of images as the two ensembles before and after the change and return the locations where the changes are likely to be happened [1]. Before such stage, always a preprocessing step is necessary to produce two comparable images in the spatial domain. The process of *registration* aims at performing some geometrical operations on one of the images, (or both of them), to give two compatible images, in which, the pixels with the same coordinates in the two images correspond to the same physical point [7]. Many researchers have reported the impact of miss–registration on the change detection results (*e.g.*see [8]). Note that the registration operation is an *inverse problem*, trying to compensate the real transformation produced by the imaging conditions. This problem has an structural relation to the problem of *camera calibration* [9], where one is concerned with estimating the 3–D coordinates of a point from its corresponding 2–D coordinates in (at least) two different cameras. A well–known model for camera projection is the *direct linear transform* (DLT) by *Abdel–Aziz* [10]. Modelling a camera with 11 parameters, the DLT is able to compensate perspective distortions [10].

In this paper, using *principal components analysis* (PCA) we propose a new fast and effective registration method. The method uses geometrical cues about the set of control points, to estimate the registration transform in a very short time. the estimation results are then enhanced in a novel formulation using linear and quadratic optimization methods to best fit the given images. Using this method, while giving high precision our method works in realtime.

The rest of this paper is organized as follows: the geometrical initialization of the registration transform is proposed in Section II-A. Two methods for computing the more precise transformation using this initialization are stated in Section II-B and Section II-C. Section III contains the experimental results and Section IV concludes the paper.

II. Proposed Method

Assume the two images I_1 and I_2, corresponding to two different shots of a single scene in different times. The aim of the *registration* process is to find the transformation $T : [x, y] \rightarrow [x^\star, y^\star]$ such that, when applying the transform T on the image I_2, the resulting image, I_2', gets aligned on the image I_1 (also called *rectification*). Here, we use some *control points* [7] in the two images. Lets call the control points in the two images, I_1 and I_2, as $\vec{x}_i$ and $\vec{y}_i$ for $i = 1, \cdots, n$, respectively. They are chosen such that, applying the transform, T, on $\vec{x}_i$ the result lies on $\vec{y}_i$. In fact, for each i, $\vec{x}_i$ and $\vec{y}_i$ correspond to the same physical point in the two images. There are some good works on automatic choosing the control points (*e.g.*,see [11]). Thus, we assume that the control points are, (somehow), found in the two images.

A. The Geometrical Initialization

In the methodology of DLT, each camera is modelled by 11 parameters and the project of the point $\vec{X} = [x, y, z]$ on a camera is defined as $u = (a_u x + b_u y + c_u z + d_u)/(ax + by + cz + 1)$ and $v = (a_v x + b_v y + c_v z + d_v)/(ax + by + cz + 1)$ [10]. The denominator term, $\lambda = ax + by + cz + 1$, applies the effects of the destination from $\vec{X}$ to the center of the camera on the projected point coordinates [10]. In the case of space–born imagery, there are two simplifications to be applied on the DLT formulation. Firstly, the vertical distance between the camera and the subject points, z, is constant, because the camera plane is almost parallel to the subject [7]. Secondly, as the normal vector of the camera plane and the normal vector of the earth surface are almost parallel, the denominator term, λ, gets

constant for all image pixels. Thus, setting $a = \frac{1}{\lambda}a_u$, $b = \frac{1}{\lambda}b_u$, $c = \frac{1}{\lambda}(c_u z + d)$, $\tilde{a} = \frac{1}{\lambda}a_v$, $\tilde{b} = \frac{1}{\lambda}b_v$, and $\tilde{c} = \frac{1}{\lambda}(c_v z + d)$ gives the simplified linear model, defined as $u = ax + by + c$ and $v = \tilde{a}x + \tilde{b}y + \tilde{c}$, also known as the *affine* transform [7]. Note that the affine transformat can be rewritten in the matrix notation as, $\vec{u} = A\vec{x} + \vec{c}$ where $A = \begin{pmatrix} a & b \\ \tilde{a} & \tilde{b} \end{pmatrix}$ and $\vec{b} = \begin{pmatrix} c \\ \tilde{c} \end{pmatrix}$. Also, the two different transformations in the affine transform resulting in u and v, are independently parameterized, in contrast with the mixed case in the conventional DLT. While in proving the theorems we use the matrix notation, in the text we show the transformation described in the affine transformation as the pair $(t_x, t_y) = ([a, b, c], [\tilde{a}, \tilde{b}, \tilde{c}])$. Table I shows some common transforms in this notation.

TABLE I

SOME COMMON TRANSFORMS IN PAIR NOTATION.

Description	t_x	t_y
Unity	$[1, 0, 0]$	$[0, 1, 0]$
Translation by $[x_0, y_0]^T$	$[1, 0, x_0]$	$[0, 1, y_0]$
Rotation around $[0, 0]$	$[\cos\theta, -\sin\theta, 0]$	$[\sin\theta, \cos\theta, 0]$
Balanced scaling by α	$[\alpha, 0, 0]$	$[0, \alpha, 0]$
Unbal. scaling (α,β)	$[\alpha, 0, 0]$	$[0, \beta, 0]$

For two pairs of transforms, (t_x, t_y) and (t'_x, t'_y), we define the transform (t''_x, t''_y) as the result of merging (t_x, t_y) and (t'_x, t'_y), writing as $(t''_x, t''_y) = (t'_x, t'_y) \circ (t_x, t_y)$, when applying (t''_x, t''_y) on an image results the same as applying (t_x, t_y) on the original image and then applying (t'_x, t'_y) on the result. Note that in the general case $(t'_x, t'_y) \circ (t_x, t_y)$ and $(t_x, t_y) \circ (t'_x, t'_y)$ are not necessarily equivalent. Numerical derivation shows that using the transform pair notation, $(t''_x, t''_y) = (t'_x, t'_y) \circ (t_x, t_y)$ implies that, $a'' = a'a + b'\tilde{a}$, $b'' = a'b + b'\tilde{b}$, $c'' = a'c + b'\tilde{c} + c'$, $\tilde{a}'' = \tilde{a}'a + \tilde{b}'\tilde{a}$, $\tilde{b}'' = \tilde{a}'b + \tilde{b}'\tilde{b}$, and $\tilde{c}'' = \tilde{a}'c + \tilde{b}'\tilde{c} + \tilde{c}'$. Note that applying numerous transformation pairs on an image is a lengthy operation, also resulting in numerical errors, while the above merging scheme enables us to merge the transformation pairs and produce just one transformation which is applied much faster and more reliable.

Our proposed method for registering two given remotely sensed images, uses the decomposition of the transformation into three elementary transformations of *translation*, *rotation* and *un–balanced scaling* (see Figure 1). In practice, the elementary transformations are also decomposed into more transformations and then the result of merging the combination of all transforms is applied on the image.

As shown in Table I, a *balanced* scaling is an *orthogonal* transformation which multiples all distances by a constant factor, while preserving angles. In contrast, an *unbalanced* transformation manipulates both angles and distances in a more complex way. It is known from linear algebra that, for an unbalanced scaling, there exists two perpendicular vectors $\vec{v}_1$ and $\vec{v}_2$ which are magnified by factors of α_1 and α_2 and the result of the transformation on any arbitrary vector can be measured using these two vectors. Note that for an arbitrary vector $\vec{x}$ there is one and only one representation as $\vec{x} = a\vec{v}_1 + b\vec{v}_2$. As the scaling transformation ($\vec{y} = S\vec{x}$) is a linear operator we have, $S\vec{x} = a\alpha_1\vec{v}_1 + b\alpha_2\vec{v}_2$. Assuming the points $\vec{x}_i$ on the unit circle, we want to find the locus of the transformed points $\vec{y}_i = S\vec{x}_i$. Note that by changing the coordinate system to the one with the axis of $\vec{v}_1$ and $\vec{v}_2$ getting $\vec{x}'_i$ out of $\vec{x}_i$ $\vec{x}'_i$s yield a circle ($\|\vec{x}'_i\| = 1$). Note that by writing the transformation matrix S as the diagonal decomposition form of VDV^{-1}, the matrix V contains the two vectors $\vec{v}_1$ and $\vec{v}_2$ as its columns and D is a diagonal matrix with terms α_1 and α_2 on its main diagon. Note that $(V^{-1}\vec{y}_i)^T D^{-2}(V^{-1}\vec{y}_i) = \vec{x}_i^T\vec{x}_i = 1$. Thus, $\vec{y}_i$s shape up an ellipse rotated by V with diagons equal to α_1 and α_2. Consider $\vec{y} = S\vec{x}$ again with the constraint $\vec{x} = 1$. As $\vec{v}_1 \perp \vec{v}_2$, we have $a^2 + b^2 = 1$. Also, $\|S\vec{x}\|^2 = a^2\alpha_1^2 + b^2\alpha_2^2$. Substituting $b^2 = 1 - a^2$ in $\Delta = a^2\alpha_1^2 + b^2\alpha_2^2$ and computing $\frac{\partial\Delta}{\partial a^2}$ to find the extremes of Δ, yields the values of α_1^2 and α_2^2 for the vectors $\vec{x} = \vec{v}_1$ and $\vec{x} = \vec{v}_2$, respectively. Thus, for the control points $\vec{y}_i$ in the second image corresponding to the control points $\vec{x}_i$ on the unit circle in the first image, performing *principal component analysis* (PCA) gives the matrix V. Note that the above method is only applicable if we have the set of control points making a unit circle in the first image.

The geometric model we are using here is,

$$\vec{y} = R(VDV^{-1}(\vec{x} - \vec{a}) - \vec{b}) - \vec{c} \tag{1}$$

where R is the rotation matrix and VDV^{-1} is the unbalanced scaling matrix. Here the two matrices R and V are *orthonormal*, D is diagonal, the vectors $\vec{a}$ and $\vec{b}$ model the points around which the scaling and rotation take place, and $\vec{c}$ is the final translation vector. Equation (1) can be rewritten in the more simpler form of $\vec{y} = A\vec{x} + \vec{B}$ where $A = RVDV^{-1}$ and $\vec{B} = -(RVDV^{-1}\vec{a} + R\vec{b} + \vec{c})$. Consider the vectors $\vec{x}_i = \vec{x}_0 + \vec{v}_i$ in the first image, where $\vec{v}_i$ satisfies $\|\vec{v}_i\| = 1$. Assume that the vector $\vec{y}_i$ is the vector corresponding to $\vec{x}_i$ in the second image for $i = 1, \cdots, n$. Then $\vec{y}_i - \vec{y}_0$ is the result of applying the transformation on $\vec{x}_i - \vec{x}_0$. Assume the case of $\vec{v}_2 = \alpha\vec{v}_1$ and the two vectors $\vec{y}_1$ and $\vec{y}_2$ as the result of the transformation shown in (1) applied on the two vectors $\vec{x}_1 = \vec{x}_0 + \vec{v}_1$ and $\vec{x}_2 = \vec{x}_0 + \vec{v}_2$, respectively. We have, $\|\vec{y}_2 - \vec{y}_0\|/\|\vec{y}_1 - \vec{y}_0\| = \|A\vec{v}_2\|/\|A\vec{v}_1\| = \alpha$. Thus, for the arbitrary vector $\vec{x}_0$ in the first image, the length of the vector $\vec{x} - \vec{x}_0$ in the first image and the length of its image in the second image just vary by a constant multiplier, which is the scaling factor in that direction. Hence, for the set of n control points $\vec{x}_i$ and $\vec{y}_i$ in the first image and the second image, the principal directions of the set $\Phi = \{(\vec{y}_i - \vec{y}_{i_0})/\|\vec{x}_i - \vec{x}_{i_0}\| i = 1, \cdots, n, i \neq i_0\}$ for an arbitrary value of i_0 in the range $1, \cdots, n$, give the matrix V, while its eigenvalues are proportional to α_1 and α_2. By using the computed V and D, the unbalanced scaling is compensated while one needs to compensate the balanced scaling independently. Note that if the scaling is balanced, Φ makes a circle and the eigenvectors get ambiguous. We propose computing the standard deviation of the set $\tilde{\Phi} = \{\|\vec{y}_i - \vec{y}_{i_0}\|/\|\vec{x}_i - \vec{x}_{i_0}\| i = 1, \cdots, n, i \neq i_0\}$ as the criteria for the scale being balanced. Note that $\sigma_{\tilde{\Phi}} \simeq 0$ show that the scaling is balanced. Also note that for the

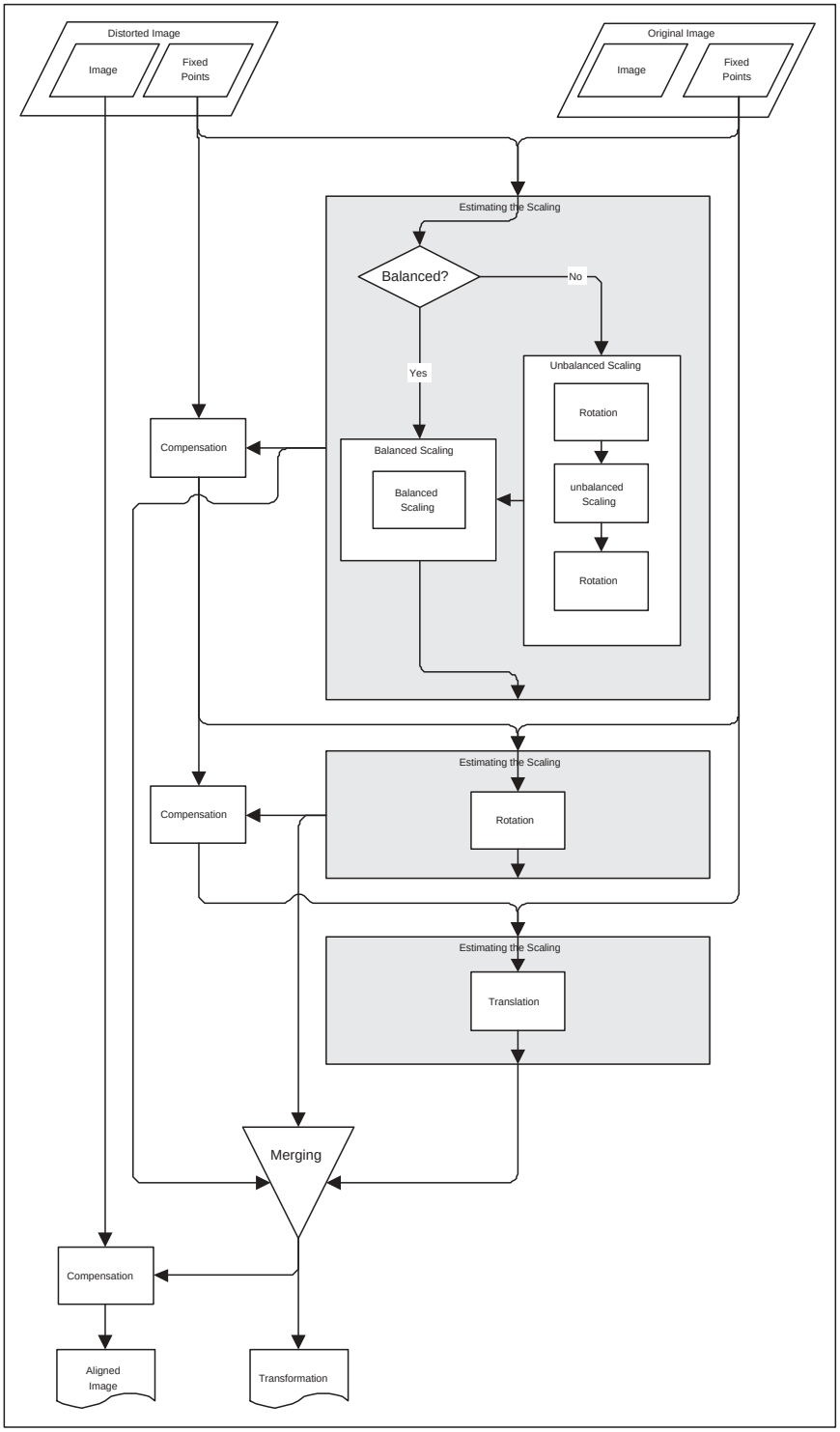


Fig. 1. Flowchart of the proposed registration method.

balanced scaling, $\eta_{\tilde{\Phi}}$ gives the scale. To find a value of i_0 which results in computing the scales in more directions, we propose using the point in the control points set, which is the nearest to the expectation vector of the control points. Note that selecting any other member of the control points set as the central point yields the same result. After estimating the scaling, we apply the produced transformation pair on the $\vec{y_i}$ to form the new control points in the second image. For simplicity we use the same $\vec{y_i}$ term for the new points. Note that in this stage the transformation is reduced to $\vec{y} = R\vec{x} + \vec{b}$,

where R is a rotation matrix. Also, note that for an arbitrary value of i_0 and $i \neq i_0$ we have $\vec{y}_i - \vec{y}_{i_0} = R(\vec{x}_i - \vec{x}_{i_0})$. The angle between two vectors in the 2–D plane is computed as $\angle(\vec{v}, \vec{u}) = sgn((\vec{v} \times \vec{u}) \cdot \vec{j}) \cos^{-1} \vec{v} \cdot \vec{u} / \|\vec{v}\| \|\vec{u}\|$, where $sgn(x)$ is the *sign* function. Now, by forming the set $\Theta = \{\angle(\vec{y}_i - \vec{y}_{i_0}, \vec{x}_i - \vec{x}_{i_0}) | i = 1, \cdots, n, i \neq i_0\}$ and computing $\eta_\Theta = E\{\theta | \theta \in \Theta\}$, the rotation angle is estimated. After estimating the rotation, we apply the produced transformation pair on $\vec{y}_i$ to form new vectors. Now, the transformation is reduced to a translation, $\vec{y} = \vec{x} + \vec{a}$. Note that $\vec{a} = E\{\vec{y}_i\} - E\{\vec{x}_i\}$. Now merging the results, the whole transformation needed to register the second image on the first image is acquired.

It is useful to show the registration result in a visual way for illustration purposes. We propose to use a color image when the underlying images are gray–scales. By putting the first image in the *red* plane and the registered version of the second image in the *green* plane while the *blue* plane is filled with 255, one can find the non–alignments pixels which will be shown as *magenta* and *cyan* regions. As a numerical criteria of matching, we propose using the normalized mean square error of the control points as the total distortion $\varepsilon = \sqrt{\sum_{i=1}^{n} \|\vec{y}_i - \vec{x}_i\|^2} / \sqrt{n(w^2 + h^2)}$, where, the first image is $h \times w$ in pixels. We call this method the geometric initialization method.

B. Gradient–Descent Enhancement

Assume the function $f : R^n \rightarrow R$. The *gradient–descent* method, is a first–order method for finding the minimum point of this function, starting from a good initialization. The standard *gradient descent* method uses $\vec{x}^\star = \vec{x} + \lambda \vec{\nabla} f$ as the next step, where λ is a user defined parameter, usually smaller than unity. We use a more controlled formula here, defined as:

$$\vec{x}^\star = \begin{cases} \vec{x} + \frac{\lambda}{\|\vec{\nabla} f\|} \vec{\nabla} f & , \|\vec{\nabla} f\| > \lambda \\ \vec{x} + \vec{\nabla} f & , \|\vec{\nabla} f\| \leq \lambda \end{cases} \tag{2}$$

where, λ is a user defined step size. This method adds more inertia to the solution.

Assume the first image I_1 on which the second image I_2 is to be registered. Also, assume that the method proposed in Section II-A has resulted in the transformation T with total distortion of ε. Note that the transform T produces the point (u, v) out of (x, y) as $u = \alpha x + \beta y + \gamma$ and $v = \alpha' x + \beta' y + \gamma'$. Assume rewriting the above transformation as,

$$\begin{cases} u = a\alpha x + b\beta y + c\gamma \\ v = a'\alpha' x + b'\beta' y + c'\gamma' \end{cases} \tag{3}$$

where, a, b, c, a', b', and c' are all equal to unity. Note that by tuning these parameters around 1, one can enhance the result. Defining,

$$\Psi(\vec{\omega}) = \sum_{i=1}^{n} (x_{ix} - a\alpha y_{ix} - b\beta y_{iy} - c\gamma)^2 + (x_{iy} - a'\alpha' y_{ix} - b'\beta' y_{iy} - c'\gamma')^2 \tag{4}$$

where $\vec{x}_i = [x_{ix}, x_{iy}]$ and $\vec{y}_i = [y_{ix}, y_{iy}]$ are the i-th corresponding control points in the first and the second image, respectively, and $\vec{\omega}$ is the vector $[a, b, c, a', b', c']$. We form the gradient vector of Ψ, as:

$$\vec{\nabla}\Psi(\vec{\omega})\Big|_{a,b,c,a',b',c'=1} = -2 \sum_{i=1}^{n} \begin{pmatrix} \alpha d_{ix} y_{ix} \\ \beta d_{ix} y_{iy} \\ \gamma d_{ix} \\ \alpha' d_{iy} y_{ix} \\ \beta' d_{iy} y_{iy} \\ \gamma' d_{iy} \end{pmatrix} \tag{5}$$

where, $d_{ix} = x_{ix} - \alpha y_{ix} - \beta y_{iy} - \gamma$ and $d_{iy} = x_{iy} - \alpha' y_{ix} - \beta' y_{iy} - \gamma'$. We propose applying (3) on (5) to minimize (5) starting from T iteratively, with the halting condition defined as ε being smaller than the preselected margin of $\tilde{\varepsilon}$ or the number of iterations becoming too much. Thus, in each iteration the transformation $T = ([\alpha, \beta, \gamma], [\alpha', \beta', \gamma'])$ changes to, $T^\star = ([\alpha(1 + \delta_\alpha), \beta(1 + \delta_\beta), \gamma(1 + \delta_\gamma)], [\alpha'(1 + \delta_{\alpha'}), \beta'(1 + \delta_{\beta'}), \gamma'(1 + \delta_{\gamma'})])$ where $[\delta_\alpha, \delta_\beta, \delta_\gamma, \delta_{\alpha'}, \delta_{\beta'}, \delta_{\gamma'}]^T$ equals $\frac{\lambda}{\|\vec{\nabla}\Psi\|} \vec{\nabla}\Psi$ if $\|\vec{\nabla}\Psi\| > \lambda \vec{\nabla}\Psi$ and $\vec{\nabla}\Psi$, otherwise. Note that all equations are written while $a, b, c, a', b', c' = 1$. Gradient–descent method is proved to go towards a minimum point of the given function, when λ is selected small enough. But by reducing λ the convergence speed declines.

C. Newton–based Enhancement

Assume the function $f : R^n \rightarrow R$. The *Newton*–based method, is a second–order method for finding an extremum point of this function, starting from a good initialization. At the extremum point we have $\frac{\partial f}{\partial x_i} = 0, i = 1, \cdots, n$ or equivalently, $\vec{\nabla} f = \vec{0}$. Defining $\vec{g}(\vec{x}) = \vec{\nabla} f|_{\vec{x}}$, gives the new function $g : R^n \rightarrow R^n$ getting its zero vector at the extremum points of f. Using the *Tailor* series expansion of $\vec{g}_i$ only to first derivative results in $g_i(\vec{x}) = g_i(\vec{x}_0) + \vec{J}(g_i)^T|_{\vec{x}_0}(\vec{x} - \vec{x}_0)$, where $\vec{J}(g_i)$ is the *Jacobian* of g_i defined as $\vec{J}(g_i) = [\frac{\partial g_i}{\partial x_1}, , \cdots, , \frac{\partial g_i}{\partial x_n}]^T$. Setting $g_i(\vec{x}) = 0$ results in $\vec{J}(g_i)^T|_{\vec{x}_0}(\vec{x} - \vec{x}_0) = -g_i(\vec{x}_0)$, which is a linear function of $\vec{dx} = \vec{x} - \vec{x}_0$. Combining the n equations, produced in this way for $i = 1, \cdots, n$, results in $H(f)(\vec{x}^\star - \vec{x}) = -g_i(\vec{x})$ or $\vec{x}^\star = \vec{x} - H(f)^{-1} g_i(\vec{x})$ as the iteration of the *Newton*–based method towards the extremum. The matrix $H(f)$ is the *Hessian* matrix of the function f. Note that the *Newton*–based method finds the local extremum (not the local minimum). Thus, proper selection of the initial point is of great importance.

Assume the first image I_1 on which the second image I_2 is to be registered. Also, assume that the method proposed in Section II-A has resulted in the transformation T with total distortion of ε. Here, we separate $\Psi(\vec{\omega})$ into two parts of $\Psi_x(\vec{\omega}_x) = \sum_{i=1}^{n} (x_{ix} - a\alpha y_{ix} - b\beta y_{iy} - c\gamma)^2$ and $\Psi_y(\vec{\omega}_y) = \sum_{i=1}^{n} (x_{iy} - a'\alpha' y_{ix} - b'\beta' y_{iy} - c'\gamma')^2$, where, $\vec{\omega}_x = [\alpha, \beta, \gamma]$ and $\vec{\omega}_y = [\alpha', \beta', \gamma']$. note that $\Phi(\vec{\omega}) = \Phi_x(\vec{\omega}_x) + \Phi_y(\vec{\omega}_y)$, $\frac{\partial}{\vec{\omega}_y} \Phi_x(\vec{\omega}_x) = \vec{0}$, and $\frac{\partial}{\vec{\omega}_x} \Phi_y(\vec{\omega}_y) = \vec{0}$. Thus, we propose minimizing Ψ_x and Ψ_y, independently. Numerical derivation shows that $\vec{J}\Psi_x(\vec{\omega}_x)|_{a,b,c=1} = -2 \sum_{i=1}^{n} [\alpha d_{ix} y_{ix}, \beta d_{ix} y_{iy}, \gamma d_{ix}]^T$ and

$$H\Psi_x(\vec{\omega}_x)|_{a,b,c=1} = 2\sum_{i=1}^{n}\begin{pmatrix} \alpha^2 y_{ix}^2 & \alpha\beta y_{ix}y_{iy} & \alpha\gamma y_{ix} \\ \alpha\beta y_{ix}y_{iy} & \beta^2 y_{iy}^2 & \beta\gamma y_{iy} \\ \alpha\gamma y_{ix} & \beta\gamma y_{iy} & \gamma^2 \end{pmatrix} \quad (6)$$

where $d_{ix} = x_{ix} - \alpha y_{ix} - \beta y_{iy} - \gamma$ and $d_{iy} = x_{iy} - \alpha' y_{ix} - \beta' y_{iy} - \gamma'$. In each iteration, the transformation $T = ([\alpha, \beta, \gamma], [\alpha', \beta', \gamma'])$ changes to, $T^{\star} = ([\alpha(1+\delta_\alpha), \beta(1+\delta_\beta), \gamma(1+\delta_\gamma)], [\alpha'(1+\delta_{\alpha'}), \beta'(1+\delta_{\beta'}), \gamma'(1+\delta_{\gamma'})])$, where, $[\delta_\alpha, \delta_\beta, \delta_\gamma]^T = -H\Psi_x^{-1}(\vec{\omega}_x)\vec{J}_x(\vec{\omega}_x)$ and $[\delta_{\alpha'}, \delta_{\beta'}, \delta_{\gamma'}]^T = -H\Psi_y^{-1}(\vec{\omega}_y)\vec{J}_y(\vec{\omega}_y)$, when setting $a, b, c = 1$ is all equations. We propose applying the above iterations with the halting condition of $|\varepsilon_{t+1} - \varepsilon_t| \leq \varepsilon_\circ$, where $\varepsilon_\circ$ is the error resolution.

III. Experimental Results

The tests were performed using a PIV 2600MHz personal computer with 512MB of RAM. Figure 2 shows some different transformations of an image and Figure 3 shows the control points in the image shown in Figure 2–(a).

(a) (b) (c) (d) (e)

Fig. 2. Different transformations of an image. (a) First image. (b) Translated. (c) Rotated and Translated. (d) Rotated, translated, and balanced scaled. (e) Rotated, translated, and unbalanced scaled.

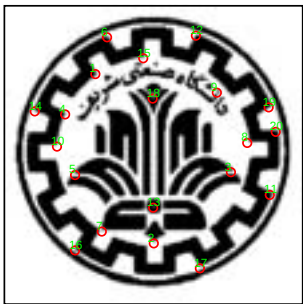
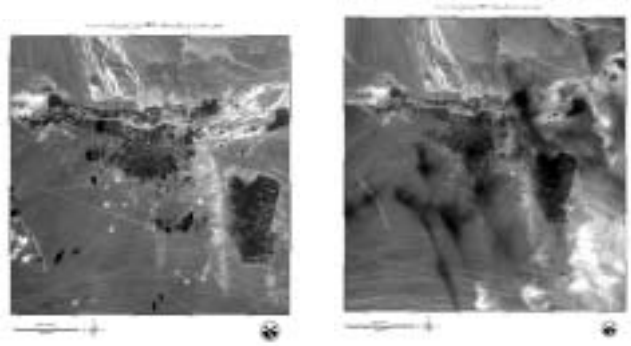


Fig. 3. Control points on the image shown in Figure 2–(a).

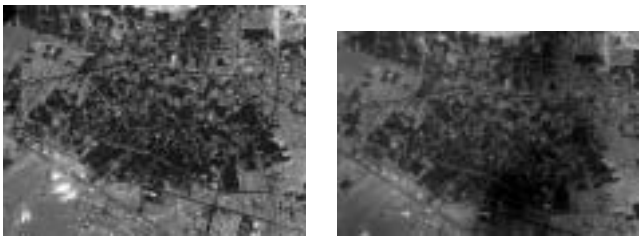
Figure 4 shows two *panchromatic* images taken from the city of *Bam* by the *IRS–1D* satellite *PAN* sensor, before and after the devastating earthquake of December 26, 2003. Figure 5 shows the urban portion of the images. The first images are cropped with no magnification to focus on details. Figure 6 shows two images of the city of *Bam* by the *IRS–1D* satellite *LIS III* sensor, each containing 3 channels.

Figure 7 shows the results of registering the images shown in Figure 2 on the image shown in Figure 2–(a) using the proposed geometric method. Here, the 20 control points shown in Figure 3 are used. Table II lists the values of total distortion and the elapsed time in this test. Table III lists the values of ε when trying to register the samples shown in Figure 2 on themselves and on each other, using the proposed geometric



(a) (b)

Fig. 4. *Panchromatic* images of the city of *Bam*. (a) 2003-12-04. (b) 2003-12-29.



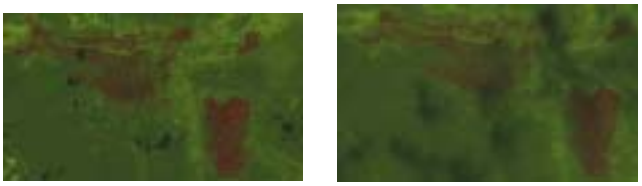
(a) (b)

Fig. 5. Urban portion of the images shown in Figure 4.

method. In this paper, the *nearest–neighborhood* interpolation method is used. As stated in [7] it is a satisfactory choice in most cases. Note that when the transformation is not massively changing the image (all except (e)), the proposed geometric registration method results in about 1% distortion. The average elapsed time in this test is $10ms$.

A test was performed to see wether twice applying the proposed geometric registration method on the second image enhances the results. In numerous images, it was observed that the value of ε and the resulting transformation are identical or negligibly different in the second application of the proposed method on the registered image.

Figure 8 shows the results of applying the proposed gradient–descent enhanced method with $\lambda = 0.002$ and $\tilde{\varepsilon} = 0.001$, ($0.1\%$ error). The maximum number of iterations is set as 10000. Table IV lists the elapsed times and total distortion values. Note the better alignment of the images, compared



(a) (b)

Fig. 6. Multi–spectral images of the city of *Bam*. (a) 2003-12-04. (b) 2003-12-29.

(a) (b) (c) (d) (e)

Fig. 7. Results of applying the proposed geometric method for registering the images shown in Figure 2 on the image shown in Figure 2–(a).

TABLE II

ELAPSED TIME AND TOTAL DISTORTION RESULTS FOR REGISTERING THE IMAGES SHOWN IN FIGURE 2 ON THE IMAGE SHOWN IN FIGURE 2–(A) USING THE PROPOSED GEOMETRIC METHOD.

2nd Image	(a)	(b)	(c)	(d)	(e)
$t(ms)$	10	11	10	10	10
ε	1.07%	1.07%	1.07%	1.06%s	1.07%

with the results of the proposed geometric method shown in Figure 7, while the elapsed time is much higher.

(a) (b) (c) (d) (e)

Fig. 8. Results of using the gradient–descent enhanced proposed method for registering the images shown in Figure 2 on the image shown in Figure 2–(a).

Figure 9 shows the results of applying the proposed *Newton*–based enhanced method with $\varepsilon_\circ = 10^{-10}$. Table V lists the elapsed times and total distortion values. Note the perfect alignment of the images, while the elapsed time is less than $20ms$ in all cases. The proposed *Newton*–based enhanced method vanishes for registering an image on itself, because the *Hessian* matrices get singular. Table VI lists the values of total distortion resulted when registering the images shown in Figure 2 on each other using the proposed *Newton*–based enhanced method. Figures 10, 11 show the results of registering the images shown in Figures 4,5, respectively, using the proposed *Newton*–based enhanced method. The resulting total distortion values are $\varepsilon = 0.011\%$ and $\varepsilon = 0.566\%$, respectively, while both has elapsed about $15ms$. Figure 12 shows the result of registering the images shown in Figure 6 with the proposed *Newton*–based enhanced method using 20 control points. The method has elapsed $16ms$ resulting in total

TABLE III

TOTAL DISTORTION RESULTS WHEN REGISTERING THE IMAGES SHOWN IN FIGURE 2 WITH EACH OTHER AND THEMSELVES USING THE PROPOSED GEOMETRIC METHOD.

1st Image / 2nd Image	(a)	(b)	(c)	(d)	(e)
(a)	1.07%	1.08%	1.07%	0.68%	4.35%
(b)	1.07%	1.07%	1.07%	0.67%	4.35%
(c)	1.08%	1.08%	1.07%	0.68%	4.35%
(d)	1.06%	1.07%	1.06%	0.66%	4.37%
(e)	1.07%	1.07%	1.07%	0.67%	4.35%

TABLE IV

ELAPSED TIME FOR REGISTERING IMAGES SHOWN IN FIGURE 2 ON THE IMAGE SHOWN IN FIGURE 2–(A) USING THE PROPOSED GRADIENT–DESCENT ENHANCED METHOD.

2nd Image	(a)	(b)	(c)	(d)	(e)
t	5.2s	5.4s	0.1s	0.1s	2.8s
ε	0.36%	0.29%	0.099%	0.099%s	0.1%

distortion of 0.10%.

(b) (c) (d) (e)

Fig. 9. Results of using the proposed method for registering the images shown in Figure 2 on the image shown in Figure 2–(a) using the proposed *Newton*–based enhanced method.

TABLE V

ELAPSED TIME FOR REGISTERING IMAGES SHOWN IN FIGURE 2 ON THE IMAGE SHOWN IN FIGURE 2–(A) USING THE PROPOSED *Newton*–BASED ENHANCED METHOD.

2nd Image	(b)	(c)	(d)	(e)
$t(ms)$	16	15	16	17
ε	0.02%	0.05%	0.05%	0.06%

TABLE VI

TOTAL DISTORTION RESULTS WHEN REGISTERING THE IMAGES SHOWN IN FIGURE 2 WITH EACH OTHER AND THEMSELVES USING THE PROPOSED *Newton*–BASED ENHANCED METHOD.

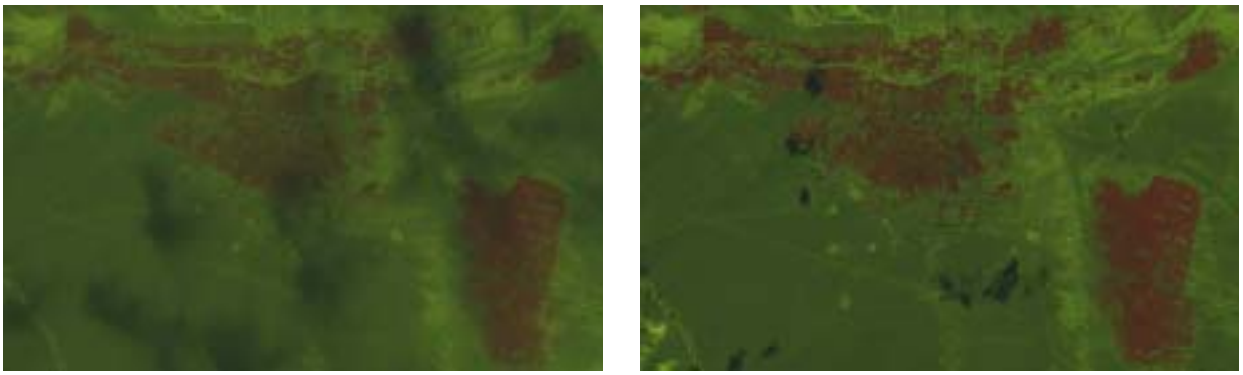
1st Image / 2nd Image	(a)	(b)	(c)	(d)	(e)
(a)	–	0.017%	0.046%	0.045%	0.062%
(b)	0.017%	–	0.049%	0.046%	0.061%
(c)	0.046%	0.049%	–	0.056%	0.066%
(d)	0.054%	0.055%	0.066%	–	0.088%
(e)	0.061%	0.061%	0.069%	0.070%	–

IV. CONCLUSIONS

A new fast and efficient registration method for remotely sensed images is proposed. The method uses a geometrical estimating of thetransform, enhanced with the *Newton*–based method, using a novel formulation. The experimental results show the total distortion results of less than half percent for a realtime application elapsing less than $12msec$s.

ACKNOWLEDGEMENT

We would like to appreciate the valuable discussions and suggestions made by professor *Y. Kosugi* and professor *M. Nakamura* from *Tokyo Institute of technology*. We also wish to thank the *Iranian Remote Sensing Center* (IRSC) for providing us with the remote sensing images used in this paper. The first author also wishes to thank Ms. *Azadeh Yadollahi* for her encouragement and invaluable ideas.



(a) (b)

Fig. 12. Result of applying the proposed *Newton*–based enhanced registration method on the images shown in Figure 6.

Fig. 10. Result of applying the proposed *Newton*–based enhanced registration method on the images shown in figure 4.

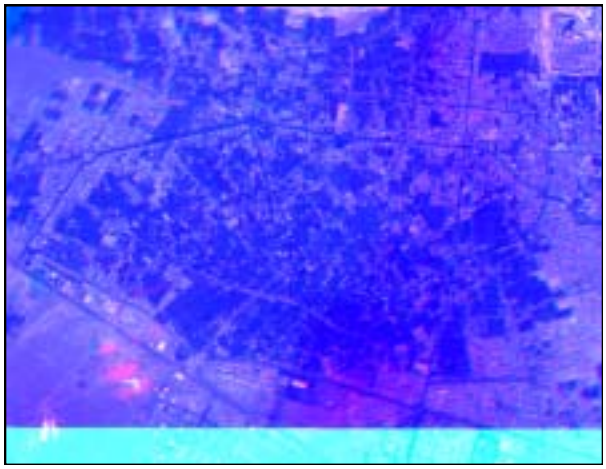


Fig. 11. Result of applying the proposed *Newton*–based enhanced registration method on the images shown in figure 5.

REFERENCES

[1] R. Wiemker, A. Spek, D. Kulbach, H. Spitzer, and H. Bienlein, "Unsupervised robust change detection on multispectral imagery using spectral and spatial features," in *Proceedings of the Third International Airborne Remote Sensing Conference and Exhibition*, Copenhagen, Denmark, 1997.

[2] C. S. Fischer and L. M. Levinen, "Monitoring california's hardwood rangelands using remotely sensed data," in *Proceedings of the Fifth Oak Symposium*, Oak Woodlands, 2001.

[3] K. M. Bergen, D. G. Brown, J. R. Rutherford, and E. J. Gustafson, "Development of a method for remote sensing of land–cover change 1980–200 in the ufs north central region using heterogeneous usgs luda and noaa avhrr 1km data," in *Proceedings, International Geoscience and Remote Sensing Symposium*, Toronto, CA, 2002.

[4] M. Matsuoka and F. Yamazaki, "Application of the damage detection method using sar intensity images to recent earthquakes," in *Proceedings of the International Geoscience and Remote Sensing Symposium, IEEE,*, 2002, pp. CD–ROM.

[5] G. Andre, L. Chiroiu, C. Mering, and F. Chopin, "Building destruction and damage assessment after earthquake using high resolution optical sensors. the case of the gujarat earthquake of 26, 2001," Unknown IEEE Conference., 2003.

[6] M. Nakamura, M. Sakamoto, S. Kakumoto, and Y. Kosugi, "Stabilizing the accuracy of change detection from geographic images by multi–levelled exploration and selective smoothing," in *Proceedings of GIS2003*, Vancouver, 2003.

[7] B. Zitova and J. Flusser, "Image registration methods: A survey," *Image and Vision Computing*, vol. 21, pp. 977–1000, 2003.

[8] J. R. G. Townshend, C. O. Justice, and C. Gurney, "The impact of misregisitration on change detection," *IEEE Transaction on Geoscience and Remote Sensing*, vol. 30(5), pp. 1054–1060, 1992.

[9] C. Chatterjee and V. P. Roychowdhury, "Algorithms for coplanar camera calibration," *Machine Vision and Applications*, vol. 12, pp. 84–97, 2000.

[10] Y. Abdel-Aziz and H. Karara, "Direct linear transformation from comparator coordinates into object space coordinates in close range photogrammetry," in *Proceedings of the ASP/UI Symposium on Close–Rage Photogrammetry*, Urbana, Illinois, 1971, pp. 1–18.

[11] H. Neemuchwala, A. Hero, , and P. Carson, "Image registration using alpha–entropy measures and entropic graphs," *European Journal on Signal Processing (http://www.elsevier.nl/locate/sigpro), Special Issue on: Content–based Visual Information Retrieval.*, 2004.