Fast Registration of Remotely–Sensed Images for Earthquake Damage Estimation

Arash Abadpour [1] and Shohreh Kasaei [2] and S. Mohsen Amiri[3]
[1] Mathematics Science Department, Sharif Univ. of Tech., Tehran, Iran, email: abadpour@math.sharif.edu
[2] Computer Engineering Department, Sharif Univ. of Tech., Tehran, Iran, email: skasaei@sharif.edu
[3] Computer Engineering Department, Sharif Univ. of Tech., Tehran, Iran, email: mo_amiri@ce.sharif.edu

***Abstract*— Analysis of the multi–spectral remotely–sensed images of the areas destroyed by an earthquake is proved to be a helpful tool for destruction assessments. The performance of such methods is highly dependant on the preprocess that registers the two shots before and after an event. In this paper, we propose a new fast and reliable change detection method for remotely–sensed images and analyze its performance. The experimental results show the efficiency of the proposed algorithm.**

***Index Terms*— Remote sensing, registration, fuzzy optimization, change detection, texture analysis.**

I. Introduction

In recent years, the spatial and the spectral resolution of the remotely sensed 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 remotely–sensed images for regular and efficient control of the environment [1], [2].

Change detection of remotely-sensed images can be viewed as a general case of a global motion estimation usually used in the video coding applications. However, the following should be noted.

- In video coding applications, objects are likely to be presented in the next frame unless we have occlusions, newly appeared objects, lightning changes, or when dealing with degraded images. But, in remote sensing applications for situations such as earthquake, we are faced with very sever situations in which large areas are likely to be totally destroyed.
- In video coding applications, the temporal rate is about 30 frames per seconds and thus one can benefit from the existing high temporal redundancy between successive frame (when there is no shot change), while in remote sensing applications the time interval between two captured multi-band images can be considerably long resulting in a very low temporal redundancy.
- In video coding applications, the segmentation and motion estimation stages can in done in a crisp fashion, while in remote sensing applications because of the different range of changes that might exist between two shots the decisions should be made in a fuzzy fashion to take advantage of its membership style soft decisions.
- In remote sensing applications, the size and the number of the multi-spectral images are much higher than those in video sequences and thus even after dimension reduction processes we still need to have very fast algorithms.
- In remote sensing applications, due to the geometrical changes in image capturing conditions, sensor type changes, and the long interval among captured images an accurate registration process is required that plays an important role in the overall performance of any change detection or classification algorithm.

According to the above mentioned problems, the global video motion techniques might be inefficient when dealing with change detection of remote sensing applications. However, the global video motion estimation can be viewed as a special case of the proposed change detection algorithm and thus the proposed algorithm can be used for such applications as well.

A key issue in analyzing the remotely–sensed images is to detect changes on the earth's 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, a preprocessing step is necessary to produce two comparable images.

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]). The registration operation is an *inverse problem*, trying to compensate the real transformation produced by the imaging conditions. Although different registration methods are introduced and analyzed [7], [9], there is no optimal solution found yet and the problem is still an active research area [10].

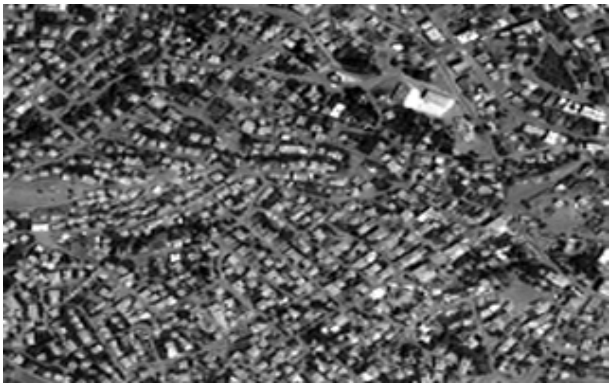
The majority of registration methods consist of four essential steps [9]:

- feature detection,
- feature matching,
- transfer model estimation, and
- image resampling and transformation.

The first step along with the second step aims at finding two

sets of corresponding points in the two images. These two sets are used in the second step to estimate the transform model. Finally, the fourth step results in the two registered images.

There are two typical methods for finding and matching feature points. The first one is to search for robust points in the two images. There are reports of using contours [11], boundaries [12], water reservoirs [13], [14], buildings [15], urban areas [16], roads [17], forests [18], costal line [19], and the forth as the features. Another approach is to use the information theory tools like *mutual information* to find the control points [20]. All of the above mentioned approaches perform both feature detection and feature matching at the same time. Due to the massive effect of mismatching of the control points on the final registration results [8], we emphasize on the determination procedure of the assigned control points (even by using the old–style approach of human intervention) for finding a set of about 20 correct control points in the two images. The challenge of using the robust control points is more clear when investigating the post–earthquake images (see Figure 1). Note that even not finding the related control points in the second image barriers valuable information about the level of occurred changes. It must be emphasized that any automatic control point detection method can be integrated to the proposed method.



(a)

(b)

Fig. 1. *Bingol*, *Turkey* area. (a) Before the earthquake 2002–07–15. (b) After the earthquake 2003–05–02. (©Digital Globe)

The rest of this paper is organized as follows: Section II describes the proposed method containing a discussion about the direct linear transform, the estimated affine transform, the related experimental results, and a proposed method to estimate the changes occurred on images. Section III contains the experimental results and discussions, and finally Section IV concludes the paper.

II. Proposed Method

Let images I_1 and I_2 correspond to two different captures of the same scene in different times. The aim of the *registration* stage is to find the transform $T : [x, y] \rightarrow [x^\star, y^\star]$ in the way that when applying the transform T on the image I_2, the resulting image I_2' gets aligned on the image I_1. We call the control points in the two images of I_1 and I_2 as $\vec{x}_i$ and $\vec{y}_i$ for $i = 1 \cdots n$, respectively. They are chosen so that applying the transform T on $\vec{x}_i$, the result lies on $\vec{y}_i$. In fact, $\vec{x}_i$ and $\vec{y}_i$ correspond to the same physical location captured as an image pixel. Here, we assume that the used control points are properly distributed all over the images.

A. Direct Linear Transform and Affine Transform

Registration has an structural relation to the problem of *camera calibration* [21], 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* and *Karara* [22]. Modelling a camera with 11 parameters, the DLT is able to compensate perspective distortions [22].

In the methodology of the DLT, each camera is modeled by 11 parameters and the projection of the point $\vec{p}_a = [x_a, y_a, z_a]$ on a camera is defined as [22],

$$x_b = \frac{a_u x_a + b_u y_a + c_u z_a + d_u}{a x_a + b y_a + c z_a + 1} \tag{1}$$

$$y_b = \frac{a_v x_a + b_v y_a + c_v z_a + d_v}{a x_a + b y_a + c z_a + 1}. \tag{2}$$

Here, the denominator term ($\lambda = ax + by + cz + 1$), applies the effects of the destination from $\vec{p}$ to the center of the camera on the projected point coordinates [22]. 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 assumed to be constant (because the camera plane is almost parallel to the subject [9]). Secondly, as the normal vector of the camera plane and the normal vector of the "on the earth"'s surface are almost parallel, the denominator term, λ, gets constant for all image pixels. Thus, setting,

$$a_1 = \frac{1}{\lambda} a_u, a_2 = \frac{1}{\lambda} b_u, t_x = \frac{1}{\lambda}(c_u z + d), \tag{3}$$

$$a_3 = \frac{1}{\lambda} a_v, a_4 = \frac{1}{\lambda} b_v, t_y = \frac{1}{\lambda}(c_v z + d), \tag{4}$$

gives the simplified linear model of,

$$x_b = a_1 x_a + a_2 y_a + t_x, \tag{5}$$

$$y_b = a_3 x_a + a_4 y_a + t_y, \tag{6}$$

also known as the *affine* transform [9]. The affine transform can be written in the matrix notation as,

$$\vec{p}_b = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \vec{p}_a + \begin{pmatrix} t_x \\ t_y \end{pmatrix}. \quad (7)$$

Note that in contrast to the conventional DLT, here the two different parts of the affine transform (that result in determining the x_b and y_b parameters) can be solved independently resulting in fastening the algorithm efficiently.

The proposed algorithm for estimating the affine transform from CPs is based on the least square error minimization approach.

1) Least Square Method: The quality of an affine transform can be measured by $Err = \sum_{i=1}^{N} \|\tilde{p}_{b,i} - p_{b,i}\|^2$. To minimize the transformation error we have to set $\nabla Err = 0$ as,

$$\begin{bmatrix} \frac{\partial Err}{\partial a_1} \\ \frac{\partial Err}{\partial a_2} \\ \frac{\partial Err}{\partial a_3} \\ \frac{\partial Err}{\partial a_4} \\ \frac{\partial Err}{\partial t_x} \\ \frac{\partial Err}{\partial t_y} \end{bmatrix} = \vec{0}. \quad (8)$$

We can rewrite Equation (8) as,

$$\begin{aligned} &a_1 \sum_{i=1}^{N} x_{a,i}^2 + a_2 \sum_{i=1}^{N} x_{a,i}.y_{a,i} + t_x \sum_{i=1}^{N} x_{a,i} = \\ &\sum_{i=1}^{N} x_{b,i}.x_{a,i}, \\ &a_1 \sum_{i=1}^{N} x_{a,i}.y_{a,i} + a_2 \sum_{i=1}^{N} y_{a,i}^2 + t_x \sum_{i=1}^{N} y_{a,i} = \\ &\sum_{i=1}^{N} x_{b,i}.y_{a,i}, and \\ &a_1 \sum_{i=1}^{N} x_{a,i} + a_2 \sum_{i=1}^{N} y_{a,i} + t_x.N = \sum_{i=1}^{N} x_{b,i} \end{aligned} \quad (9)$$

$$\begin{aligned} &a_3 \sum_{i=1}^{N} x_{a,i}^2 + a_4 \sum_{i=1}^{N} x_{a,i}.y_{a,i} + t_y \sum_{i=1}^{N} x_{a,i} = \\ &\sum_{i=1}^{N} y_{b,i}.x_{a,i}, \\ &a_3 \sum_{i=1}^{N} x_{a,i}.y_{a,i} + a_4 \sum_{i=1}^{N} y_{a,i}^2 + t_y \sum_{i=1}^{N} y_{a,i} = \\ &\sum_{i=1}^{N} y_{b,i}.y_{a,i}, and \\ &a_3 \sum_{i=1}^{N} x_{a,i} + a_4 \sum_{i=1}^{N} y_{a,i} + t_y.N = \sum_{i=1}^{N} y_{b,i} \end{aligned} \quad (10)$$

Now, using this derivation we just need to solve two linear equations of order three simultaneously. Note that, the computational complexity order of the proposed algorithm has reduced to only $O(N)$ instead of conventional approach that is in order of $O(N^3)$.

2) Experimental Results: The performance of the proposed algorithm is analyzed in terms of its complexity and accuracy. To implement the algorithm, we have used Matlab 6.5 on a 1.7 GHz, Intel Pentium M computer with 512 MB of RAM. The accuracy of different algorithms to approximate the affine transform between two sets of CPs and the related error caused during the processes are listed in Table I. The error is calculated using,

$$Error = \frac{1}{N} \frac{1}{\sqrt{W^2 + H^2}} \sum_{i=1}^{N} \left| \vec{p}_{b,i} - \left(A\vec{p}_{a,i} + \vec{t} \right) \right| \quad (11)$$

where w and h denote the width and height of the image, respectively. Table II lists the computational cost when using different number of CPs. (The common number of CPs depends on the application but an appropriate value is a number between $20 - 30$.)

As the registration step plays an important role in the overall performance of any change detection approach, and the remotely-sensed images cannot well illustrate the accurate performance of the proposed registration algorithm, here we have used a sample image (the logo of our university) to better illustrate the accurate performance of the proposed registration method. In Figure 3 we have shown different transforms

Fig. 2. A sample image.

applied on the logo images shown in Figure 2. Figure 4 shows the logo image with a set of control points overlaid on it. Figure 5 shows the result of performing our estimated affine transform on the transferred images shown in Figure 3. Here, we have used a new visualization method in which we have put the two registered images in the *red* and *green* color channels of an image and have filled the *blue* color channel with 255 value. As such, the magenta and *cyan* pixels clearly show the mis-registered locations. Note that in this figure the pixels with *cyan* colors are resulted from the borders of the transformed images shown in Figure 3 and not because of any inaccuracy in the proposed registration method. To further illustrate the results we have shown some mis-registered images in Figure **??**.

TABLE I

PERFORMANCE OF DIFFERENT ALGORITHMS.

Algorithm	Run Time	Error	Stability
Gradient-Descent[23]	2700ms	18.96%	No
Geometric[23]	10ms	1.07%	Yes
Enhanced Geometric[23]	16ms	0.045%	Yes
Fourier Transform[24]	3.8ms	0.027%	Yes
Proposed LMS	0.5 ms	0.010%	Yes

TABLE II

REQUIRED RUN TIME WHEN USING DIFFERENT NUMBER OF CONTROL POINTS.

Number of CPs	N = 10	N = 20	N = 100	N = 200
Fourier Transform[24]	1.06ms	3.8ms	108.95ms	445ms
Proposed LMS	0.34ms	0.50ms	2.43 ms	4.72ms

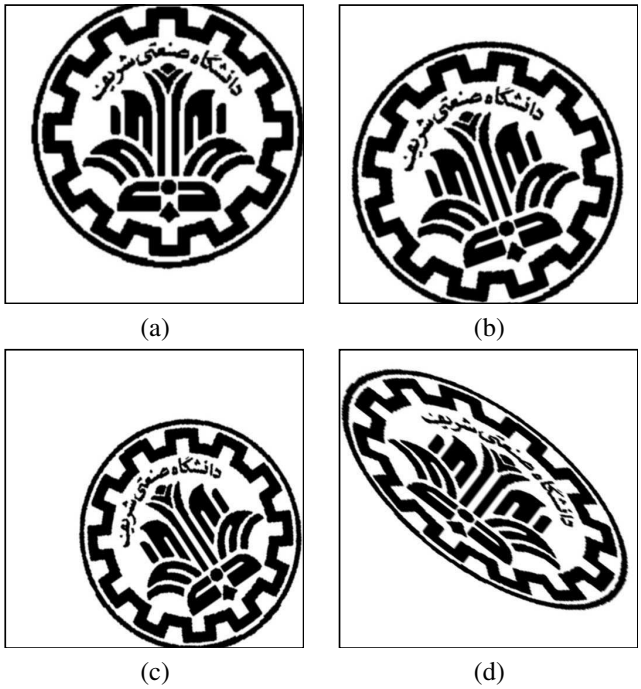


Fig. 3. Different transformations of the logo image shown in Figure 2. (a) Translated. (b) Rotated and translated. (c) Rotated, translated, and balanced scaled. (d) Rotated, translated, and unbalanced scaled.

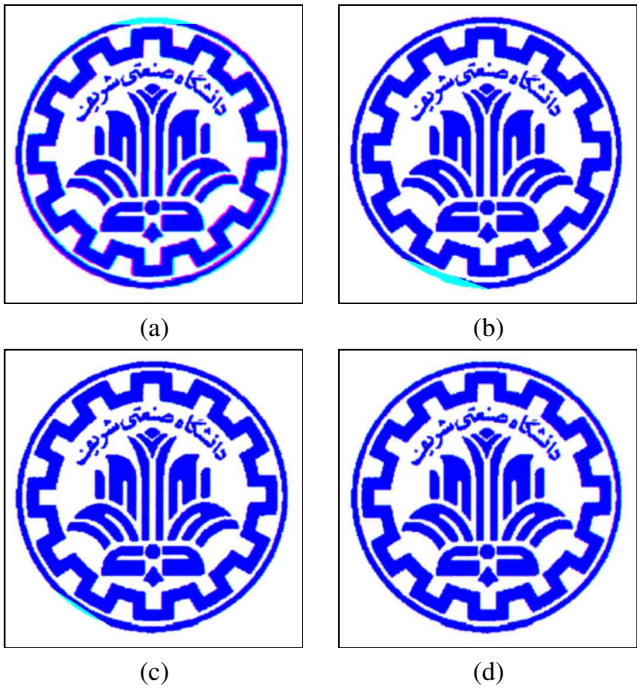


Fig. 5. Results of performing the proposed estimated affine transform on the transformed images shown in Figure 3.

Fig. 4. Control points overlaid on the logo image shown in Figure 2.

B. Proposed Change Detection Method

In this section, we state our proposed unsupervised method for segmentation and change detection in multi-spectral remotely–sensed image intervals using the proposed fuzzy principal component analysis–based clustering method. While the proposed method is faster than the available approaches reported in the literature, and depends on no predetermined parameters, it is also robust against illumination changes. To the best knowledge of the authors the method introduced in this paper is the first fuzzy change detection process. Note that the proposed affine transform estimation and the proposed change detection methods can also be used in other applications such as video motion estimation.

The literature of multi-spectral segmentation is not so rich compared to the case of grayscale segmentation methods. The first significant method for measuring the color–based similarity between two images might be the color histogram intersection approach intoduced by *Swain* and *Ballard* [25]. Although, the method is very simple, it gives a relatively reasonable performance with two main shortcomings: the lack of spatial information about the images, and dependency to imaging conditions (like the ambient illumination). Some other researchers try to use certain color spaces that they believed to be suitable for segmentation purposes. For example in [26] the authors use a geometrical measure in the color histogram to define the similarity between color pairs in the HLS color space. Although, some good segmentation results in the HLS color space are reported [27], it is proved in various studies that none of the standard color spaces are outperforming the others (*e.g.* see [28], [29]), while the local *principal component analysis* (PCA) is proved to give dominantly better results [29], [30]. In [31], the researchers process color components independently, neglecting the vector tendency of them. In [32] motion estimation is used for segmentation purposes. Here, we used all $m--D$ data in our proposed PCA–based clustering and change detection stages.

Let two images I_1 and I_2 to belong to the same scene. Then, each pixel in I_1 and I_2 is an m–D realization. Also, let image I_1 to be segmented into c classes of ϕ_i using the proposed FPCAC method [33]. Here, J_{ixy} shows the membership of $\vec{I}_{1xy}$ to the i–th class.

Now, perform the FPCA [33] on the fuzzy set,

$$\tilde{X} = \{(\vec{I}_{2xy}; J_{ixy}^m) | 1 \leq x \leq W, 1 \leq y \leq H\}, \qquad (12)$$

to find the new clusters $\tilde{\phi}_i$. In fact, we are using the temporal redundancy of successive images, assuming that the fuzzy membership of a pixel to the c classes remains constant if there is no abrupt change. The reason behind finding the new clusters in I_2 is to compensate probable slight changes corresponding to the lighting and sensor changes. Now, we

have the new membership values $\tilde{J}_{ixy}$, which show the level of membership of $\vec{I}_{2xy}$ to the i–th new class $\tilde{\phi}_i$.

We propose computing,

$$\delta_{xy}^2 = \frac{1}{c^2}\sum_{i=1}^{c} J_{ixy}(J_{ixy} - \tilde{J}_{ixy})^2, 1 \le x \le W, 1 \le y \le H, \quad (13)$$

as the probability of the point (x, y) being changed from I_1 to I_2. In fact, δ_{xy} measures the net amount of change in membership of pixels to the classes in the successive images. Note that while these fuzzy change values are computed, the clusters are also updated at the same time.

If $I_1 \equiv I_2$, then J_{ixy} and $\tilde{J}_{ixy}$ will be identical, resulting in δ_{xy} being zero everywhere, as desired. Now, assume that there is no change between the two images I_1 and I_2, unless for the changes in the imaging conditions. Assume that $\vec{x}_i$ and $\vec{y}_i$ are the spectral vectors of the same pixel in the two images I_1 and I_2, respectively. We model the change in imaging conditions as a linear operation [34]. Assume that $\vec{x}_i$ and $\vec{y}_i$ relate through a linear transform namely $\vec{x}_i = A\vec{y}_i + \vec{b}$. Here, we model A as a non–singular invertible matrix with its eigenvalues being almost constant. This situation relates to the cases that the spectral axes rotate (changing the chromaticity of the illumination), scale (changing the achromaticity of the illumination), and translate. The model restricts unbalanced scaling of spectral components which changes the spectral information non–meaningfully (for details see [34]). Note that matrix A in the *singular value decomposition* (SVD) form is written as $A = VDU^{-1}$. Where, U and V are orthogonal matrices and D is a diagonal matrix with the eigenvalues of A as its elements.

The expectation vectors in the two images I_1 and I_2 relate as $E\{\vec{x}_i\} = E\{A\vec{y}_i + \vec{b}\} = AE\{\vec{y}_i\} + \vec{b}$. The fuzzy covariance matrices of the two images I_1 and I_2 satisfy $C_1 = AE\{(\vec{y}_i - E\{\vec{y}_j\})(\vec{y}_i - E\{\vec{y}_j\})^T\}A^T = AC_2A^T$. Assume that the eigenvectors of C_1 are $\vec{v}_i$ corresponding to the eigenvalues of λ_i and the eigenvectors of C_2 are $\vec{u}_i$ corresponding to the eigenvalues of ρ_i. Also, assume the eigenvectors of A to be $\vec{w}_i$ corresponding to the eigenvalues of ε_i. Thus, for all i, $C_1\vec{v}_i = \lambda_i\vec{v}_i$, $C_2\vec{u}_i = \rho_i\vec{u}_i$, and $A\vec{w}_i = \varepsilon_i\vec{w}_i$. First assume that the eigenvectors of A are all exactly equal to the fixed value of λ (or equivalently $\forall i, \varepsilon_i = \lambda$). Thus, $A = VDU^{-1}$ equals $Vdiag(\lambda, \cdots, \lambda)U^{-1} = \lambda VU^{-1}$. In this situation, $A^T = \lambda UV^{-1} = \lambda^2 A^{-1}$ resulting in $A^TA = AA^T = \lambda^2 I$. Now, note that $C_1A\vec{u}_i = AC_2A^TA\vec{u}_i = \lambda^2 AC_2\vec{u}_i = \lambda^2\rho_i A\vec{u}_i$. Thus, $A\vec{u}_i$ is the eigenvector of C_2 corresponding to the eigenvalue of $\lambda^2\rho_i$. Note that, $\|A\vec{u}_i\| = \lambda\|\vec{u}_i\| = \lambda$. As the eigenvalues and eigenvectors of a single matrix are identical, $\{(\frac{1}{\lambda}A\vec{u}_1, \lambda^2\rho_1), \cdots, (\frac{1}{\lambda}A\vec{u}_m, \lambda^2\rho_m)\}$ is identical to $\{(\vec{v}_1, \lambda_1), \cdots, (\vec{v}_m, \lambda_m)\}$. As $\lambda^2 > 0$ we have $\vec{v}_i = \frac{1}{\lambda}A\vec{u}_i$ and $\lambda_i = \lambda^2\rho_i$, for all i. Thus, using the above re–clustering method, the cluster $\phi = [\vec{\eta}, \vec{v}]$ in I_2 results in the cluster $\tilde{\phi} = [A\vec{\eta} + \vec{b}, A\vec{v}]$. Now, we have,

$$\Psi(\vec{x}_i, \tilde{\phi}) = \|[(A\vec{y}_i + \vec{b}) - (A\vec{\eta} + \vec{b})] - \quad (14)$$
$$\frac{1}{\lambda^2}\vec{v}^T A^T[(A\vec{y}_i + \vec{b}) - (A\vec{\eta} + \vec{b})]A\vec{v}\|^2 = \lambda^2\Psi(\vec{x}_i, \tilde{\phi}),$$

and $\tilde{J}_{ixy} = J_{ixy}$, resulting in $\delta_{xy} = 0$. Thus, the proposed method will be independent of the lighting and imaging conditions. Now, assume a more realistic case that ε_is are not exactly the same but we have $\lambda - \delta\lambda \le \varepsilon_i \le \lambda + \delta\lambda$. For the cases that $\frac{\delta\lambda}{\lambda}$ is too small the above equations change to semi–equations and still marginally hold. In this situation $\delta_{xy} \simeq 0$.

In contrast, physical changes of interest result in different materials in a single point in different shots. Hence, they produce absolutely different values of J_{ixy} and $\tilde{J}_{ixy}$, resulting in non–zero patterns of δ_{xy}. In the proposed method, at the same time both the image sequence segmentation and the fuzzy change detection are performed.

III. Experimental Results

The experiments are performed using an Intel Centrino 1700 MHz computer with 512 MB of RAM.

Figure 6 shows two multi–band images taken from the city of *Bam* by the *Quick Bird* satellite, before and after the devastating earthquake of December 26, 2003 before registration. Figure 7 shows the result of our registration. Figure 8 shows the urban portion of the images. The first images are cropped with no magnification to focus on details.

Figure 9 shows the resulted fuzzy change maps and a crisp map can be easily generated after performing a hard threshold.

As mentioned before, the proposed algorithm computes both the segmentation and the change detection map at the same time. Note that many applications need to use them at same time. Figure 10 illustrates the segmentation result before the earthquake and the segmentation tuning results after the earthquake.

To show the robustness of the proposed algorithm against changes in imaging conditions we have evaluated its change detection performance when running it of two images with manipulated color changes. In fact, Figure 11 shows a simulated change in imaging conditions with no real changes on the earth surface. Figures 12 and 13 illustrate the robustness of the proposed algorithm against such changes. Here, we chose a linear transform with eigenvalues 0.9, 0.7, 0.9, which are not completely equal to simulate the more realistic changes. When running the proposed change detection stage on 472×792 downsampled images it elapsed 5.7 seconds.

IV. Conclusion

In this paper, a fast and accurate affine transform estimation method and a new efficient fuzzy change detection method are proposed for remotely–sensed images. The experimental results show that the proposed method is fast and robust against undesired change in imaging conditions. It was shown that the algorithm can also efficiently used to detect damages caused by earthquake.

Acknowledgement

This work was in part supported by a grant from ITRC. We would like to appreciate the valuable discussions and suggestions made by professor *M. Nakamura* and professor *Y. Kosugi* from *Tokyo Institute of Technology*. We also wish to thank the *Iranian Remote Sensing Center* (IRSC) and *Digital Globe* for providing us with the remotely–sensed images used

(a)

(b)

Fig. 6. *Bam* area. (a) Unregistered image before the earthquake 2003-12-04. (b) Unregistered image after the earthquake 2003-12-29. (©Digital Globe)

(a)

(b)

Fig. 7. *Bam* area. (a) Registered image before the earthquake 2003-12-04. (b) Registered image after the earthquake 2003-12-29.

in this paper. Arash Abadpour also wishes to thank Ms. *Azadeh Yadollahi* for her encouragement and invaluable ideas.

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.

[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," International Geoscience and Remote Sensing Symposium, 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] L. Brown, "A survey of image registration techniques," *ACM Computing Surveys*, vol. 24 No. 4, pp. 325–376, 1992.

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

[10] D. Robinson and P. Milanfar, "Fundamental performance limits in image registration," in *IEEE International Conference on Image Processing (ICIP'03)*, 2003.

[11] H. Li, B. S. Manjunath, and S. K. Mitra, "A contour–based approach to multisensor image registration," *IEEE Transactions on Image Processing*, vol. 4 No. 3, pp. 320–333, 1995.

[12] M. Xia and B. Liu, "Image registration by "super–curves"," *IEEE Transactions on Image Processing*, vol. 13 No. 5, pp. 720–732, 2004.

[13] M. Holm, "Towards automatic rectification of satellite images using feature based matching," in *Proceedings of the International Geoscience and Remote Sensing Symposium IGARSS91*, Espoo, Finland, 1991, pp. 2439–2442.

[14] A. Goshtasby and G. Stockman, "Point pattern matching using convex hull edges," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, pp. 631–637, 1985.

[15] Y. Hsieh, D. McKeown, and F. Perlant, "Performance evaluation of scene registration and stereo matching for cartographic feature extraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 214–237, 1992.

[16] M. Roux, "Automatic registration of spot images and digitized maps," in *Proceedings of the IEEE International Conference on Image Processing ICIP96*, Lausanne, Switzerland, 1996, pp. 625–628.

[17] S. Li and J. K. M. Petrou, "Matching and recognition of road networks from aerial images," in *Proceedings of the Second European Conference on Computer Vision ECCV92*, St Margherita, Italy, 1992, pp. 857–861.

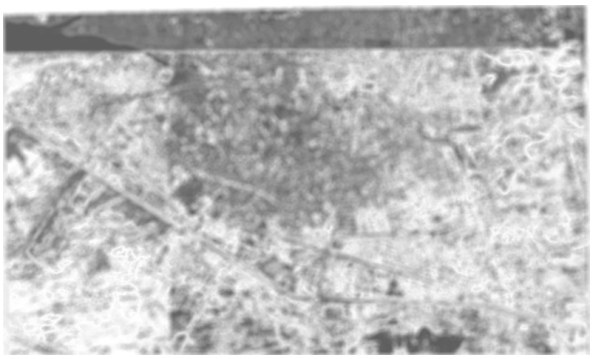
[18] M. Sester, H. Hild, and D. Fritsch, "Definition of ground control features for image registration using gis data," in *Proceedings of the Symposium*

on Object Recognition and Scene Classification from Multispectral and Multisensor Pixels, Columbus, Ohio, 1998.
[19] H. Maitre and Y. Wu, "Improving dynamic programming to solve image registration," *Pattern Recognition*, vol. 20, pp. 443–462, 1987.
[20] H. Neemuchwala, A. Hero, , and P. Carson, "Image registration using alpha–entropy measures and entropic graphs," *European Journal on Signal Processing , Special Issue on: Content–based Visual Information Retrieval.*, 2004.
[21] C. Chatterjee and V. P. Roychowdhury, "Algorithms for coplanar camera calibration," *Machine Vision and Applications*, vol. 12, pp. 84–97, 2000.
[22] 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.
[23] A. Abadpour and S. Kasaei, "Fast registration of remotly sensed images,," *10th Annual Conference of Computer Society of Iran*, pp. 61–67, 2005.
[24] S. M. Amiri and S. Kasaei, "An ultra fast method to approximate affine transform using control points," *IEEE Signal Processing Letters*, 2005.
[25] M. Swain and D. Ballard, "Color indexing." *International Journal of Computer Vision*, vol. 7(1), pp. 11–32, 1991.
[26] H. Y. Lee, H. H. Lee, and Y. H. Ha, "Spatial color descriptor for image retrieval and video segmentation," *IEEE Transaction on Multimedia*, vol. 5(3), pp. 358–367, 2003.
[27] D. Androutsos, K. Plataniotis, and A. Venetanopoulos, "Efficient indexing and retrieval of color image data using a vector–based approach," Ph.D. dissertation, University of Toronto, Toronto, ON, Canada, 1999.
[28] M. C. Shin, K. I. Chang, and L. V. Tsap, "Does color space transformation make any difference on skin detection?" in *IEEE Workshop on Applications of Computer Vision*, Orlando, FL, 2002, pp. 275–279.
[29] A. Abadpour and S. Kasaei, "A new parametric linear adaptive color space and its pca–based implementation," in *The 9th Annual CSI Computer Conference, CSICC*, Tehran, Iran, 2004, pp. 125–132.
[30] ——, "Performance analysis of three homogeneity criteria for color image processing," in *IPM Workshop on Computer Vision*, Tehran, Iran, 2004.
[31] Y. Du, C.-I. Chang, and P. D. Thouin, "Unsupervised approach to color video thresholding," *Optical Engineering*, vol. 43 (2), pp. 282–289, 2004.
[32] I. Patras, E. A. Hendriks, and R. L. Lagendijk, "Semi–automatic object–based video segmentation with labelling of color segments," *Signal Processing: Image Communication*, vol. 18, pp. 51–65, 2003.
[33] A. Abadpour and S. Kasaei, "A new fpca–based fast segmentation method for color images," in *The 4th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT 2004)*, Rome, Italy, 2004.
[34] D. O. Nikolaev and P. O. Nikolayev, "Linear color segmentation and its implementation," *Computer Vision and Image Understanding*, vol. 94, pp. 115–139, 2004.

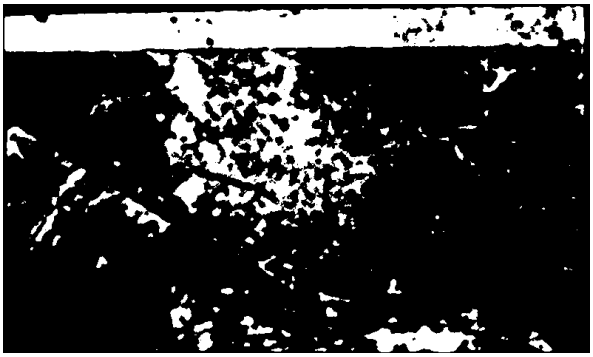
(a)

(a)

Fig. 8. Urban portion of the images shown in Figure 7.

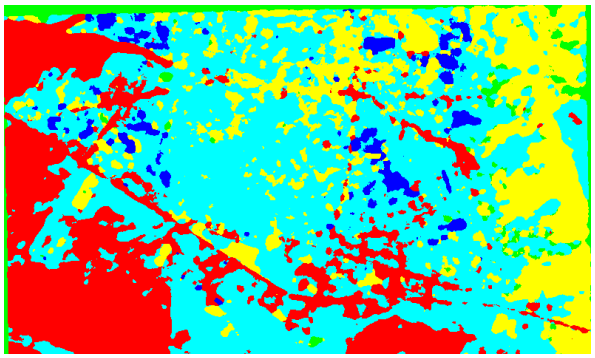


(a)

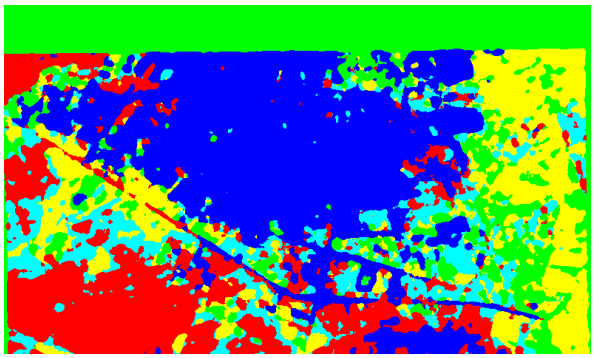


(b)

Fig. 9. Resulting change maps using the proposed change detection algorithm. (a) Fuzzy change map. (b) Crisp change map (after hard thresholding).



(a)

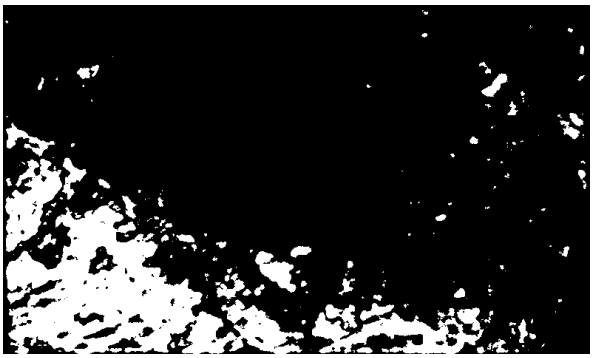


(b)

Fig. 10. Segmentation results. (a) Before the earthquake. (b)Segmentation tuning after the earthquake.

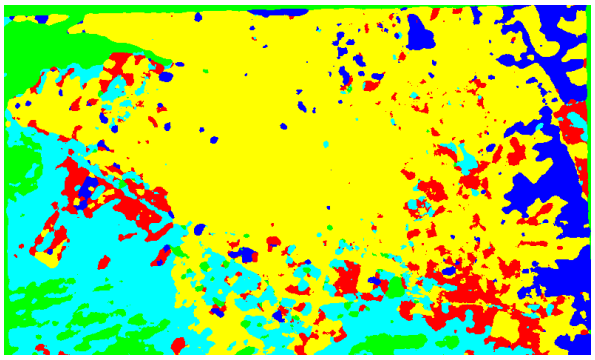
Fig. 11. Linearly changed image.

(a)

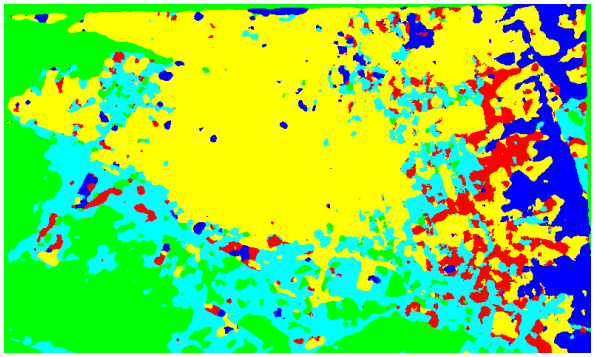


(b)

Fig. 12. Resulting change maps using the proposed change detection method (Linearly changed image). (a) Fuzzy change map. (b)Crisp change map (after hard thresholding).



(a)



(b)

Fig. 13. Segmentation results. (a) Original image. (b) Linearly changed image.