An Image Registration Method based on the Local and Global Structures

Nan Peng[1], Zhiyong Huang[1, 2], Zujun Hou[2]

[1] School of Computing, National University of Singapore
[2] Institute for Infocomm Research (I^2R), Singapore
huangzy@comp.nus.edu.sg or zyhuang@i2r.a-star.edu.sg

Abstract. We propose a novel feature-based image registration method using both the local and global structures of the feature points. To address various imaging conditions, we improve the local structure matching method. Compared to the conventional feature-based image registration methods, our method is robust by guaranteeing the high reliable feature points to be selected and used in the registration process. We have successfully applied our method to images of different conditions.

Keywords: Multimedia Content Analysis, Multimedia Signal Processing, Image Registration

1 Introduction

Image registration, an important operation of multimedia systems, is a process of transforming the different images of the same scene taken at different time, from different view points, or by different sensors, into one coordinate system. The current automated registration techniques can be classified into two broad categories: area-based and feature-based [1, 5].

In this paper, we propose and implement a novel image registration method to improve the quality of registration by guaranteeing the high reliable feature points to be selected and used in the registration process. Here we adapt the feature matching method proposed by Jiang and Yau [4]. However, it is mainly for fingerprint image under rotation and translation. We modify it so that we can obtain a set a reliable corresponding feature points for images of various conditions. The major contributions are: (1) we improve the quality of registration by applying a more reliable feature point selection and matching algorithm adapted from finger print matching, (2) we improve the local structure matching method, and (3) we implement the method in a software system and conduct various experiments with good results.

2 Our work

In this section, we describe how to extract the feature points and estimate their orientation (2.1), find correct matching pairs between two partially overlapping images (2.2), and derive the correct transformation between two images (2.3).

2.1 Feature point detection and orientation estimation

In our approach, the features are defined as points of large eigenvalues in the image. We employ the OpenCV function GoodFeaturetoTrack [3]. A number of methods have been proposed for orientation estimation of the feature points. We apply the least mean square estimation algorithm. A feature point is eliminated if its reliability of the orientation field is below a threshold.

2.2 Feature point matching

There are four major steps in our matching algorithm: an invariant feature descriptor to describe the local positional relations between two feature points (2.2.1), local (2.2.2) and global (2.2.3) structure matching, and cross validation to eliminate the false matching pairs (2.2.4). In (2.2.2), we describe our improvement.

2.2.1 Define a feature descriptor

We first represent each feature point i detected by a feature vector f_i as:

$$f_i=(x_i,y_i,\varphi_i), \tag{1}$$

where (x_i,y_i) is its coordinate, φ_i is the orientation. The feature vector f_i represent a feature point's global structure. A feature descriptor F_{ij} is defined to describe the local positional relations between two feature points f_i and f_j by their relative distance d_{ij}, radial angle θ_{ij} and orientation difference φ_{ij} (see Fig. 1) as equation (2):

$$F_{ij}=\begin{bmatrix} d_{ij} \\ \theta_{ij} \\ \varphi_{ij} \end{bmatrix}=\begin{bmatrix} \sqrt{(x_i-x_j)^2+(y_i-y_j)^2} \\ d\phi(\arctan\dfrac{y_i-y_j}{x_i-x_j},\varphi_i) \\ d\phi(\varphi_i,\varphi_j) \end{bmatrix}, \tag{2}$$

where $d\phi(t_1,t_2)$ is a function to compute the difference between two angles t_1 and t_2,. All these terms are shown in Fig. 1 for two feature points.

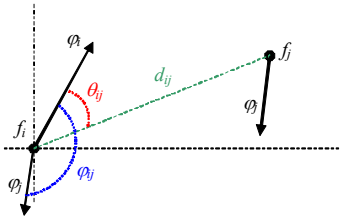


Fig. 1. The local spatial relation between two feature points f_i and f_j.

2.2.2 Local structure matching

Employing the feature descriptor, for every feature point f_i, a local structure LS_i is formed as the spatial relations between the feature point f_i and its k-nearest neighbors:

$$LS_i=(F_{i1}, F_{i2}, \ldots, F_{ik}), \tag{3}$$

where F_{ij} is the feature descriptor consisting of the local positional relations between two minutiae f_i and f_j, as defined in equation (1).

Given two feature sets $\boldsymbol{F}_s=\{f_{s1}, \ldots f_{sn}\}$ and $\boldsymbol{F}_t=\{f_{t1}, \ldots f_{tm}\}$ respectively, the aim is to find two best-matched local structure pairs $\{f_{sp} \leftrightarrow f_{tq}\}$ and $\{f_{su} \leftrightarrow f_{tv}\}$ to serve as the corresponding reference pair later in the global matching stage.

Now, we start to describe direct local structure matching [4] and complex local structure matching (the proposed improvement).

Direct local structure matching

Suppose LS_i and LS_j are the local structure feature vectors of the feature points i and j from sensed image s and template image t respectively. Their similarity level is:

$$sl(i,j)=\begin{cases} bl - W\left|LS_i - LS_j\right|, & if\ W\left|LS_i - LS_j\right| < bl \\ 0, & otherwise \end{cases} \tag{4}$$

$$W=\{\underbrace{w \ldots w}_{k}\},\ where\ w=(w_d, w_\theta, w_\varphi).$$

where W is a weight vector that specifies the weight associate with each component of the feature vector. The threshold bl can be defined as a function of the number of feature points in a neighborhood. The similarity level $sl(i,j),\ 0 \le sl(i,j) \le 1$, describes a matching certain level of a local structure pair. The two best-matched local structure pairs $\{f_{sp} \leftrightarrow f_{tq}\}$ and $\{f_{su} \leftrightarrow f_{tv}\}$ is obtained by maximizing the similarity level [4]. The direct local structure matching method is efficient of $O(k)$, where k is the number of feature points in a neighborhood.

Complex local structure matching

Though the direct local structure matching method is efficient, we found that if there are any dropped or spurious feature points in the neighborhood disturbing the order, the local structure matching will be invalid. We show an example in Fig. 2 to demonstrate this case.

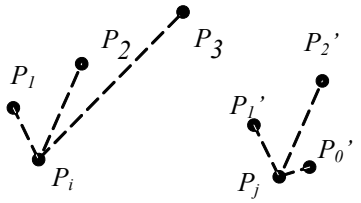


Fig. 2. Illustration of spurious or dropped feature points in the neighborhood.

In Fig. 2, p_i in the sensed image s has a neighborhood $\{p_1, p_2, p_3\}$, and p_i's corresponding point p_j in the template image t has a neighborhood $\{p_0', p_2', p_3'\}$, of which $\{p_1 \leftrightarrow p_1'\}$ and $\{p_2 \leftrightarrow p_2'\}$. Because of the image distortion or scene change, in the neighborhood of p_j, there is no matching feature point for p_3, but a spurious feature

point p_0' which does not match to any feature point in the neighborhood of p_j appears. Apply the direct local structure matching, we have $LS_i=\{F_{i1}^T,F_{i2}^T,F_{i3}^T\}$, $LS_j=\{F_{j1}^T,F_{j2}^T,F_{j3}^T\}$. Using equation (4), the similarity level between the local structures will be very low since their neighbors are mismatched. Thus the similarity level computed by equation (4) is not reliable.

We address the problem by a more complex local structure matching method. First, when we match the two neighbors of two candidate feature points, we consider not only the relative distance but also the radial angle and orientation difference. Second, after we identify those matched neighbors, we will drop the unmatched feature points in the neighborhood in computation of the similarity level of two local structures. In the example shown in Fig. 2, only $\{p_1 \leftrightarrow p_1'\}$ and $\{p_2 \leftrightarrow p_2'\}$ will be considered in the local structure matching.

Suppose we are checking the similarity level between the feature point p and q from the sensed image s and the template image t respectively. Let Knn_p and Knn_q denote the k-nearest neighborhood of the feature point p and q respectively. For every feature point n in Knn_p, we will find its most similar point m in Knn_q. They are qualified as a matching pair if three conditions (equations (5), (6) and (7)) are satisfied:

$$W|F_{pn}-F_{qm}|=min_j W|F_{pn}-F_{qj}| \text{ and } W|F_{pn}-F_{qm}|=min_i W|F_{pi}-F_{qm}|, \quad (5)$$

where $W|F_{pn}-F_{qm}|=w_d|d_{pn}-d_{qm}|+w_\theta|\theta_{pn}-\theta_{qm}|+w_\varphi|\varphi_{pn}-\varphi_{qm}|$. It searches every member in Knn_q and every member in Knn_p.

$$W|F_{pn}-F_{qm}|<T_c, \quad (6)$$

where T_c is threshold value and W is a weight vector same as in equation (4).

$$|\theta_{np}-\theta_{mq}|\leq\pi/4. \quad (7)$$

As we know if both $\{n \leftrightarrow m\}$ and $\{p \leftrightarrow q\}$ are matching pair, the relative orientation difference between θ_{np} and θ_{mq} should be small (equation (7)). Adding this criterion will speed up the search time. If the constraint is not satisfied, it is not necessary to test conditions 1 and 2.

Then the similarity level between the feature points p and q can be computed as

$$sl(p,q)=(bl-nsl(p,q))/bl, \quad (8)$$

where $nsl(p,q)=\Sigma_{n,m} W|F_{pn}-F_{qm}|$, the similarly level only for those matched neighbor pairs from Knn_p and Knn_q according to conditions 1-3. From condition 2, we have $W|F_{pn}-F_{qm}|<T_c$ if point n and point m are matched neighbors. Thus we define threshold bl as T_c times the number of matching neighbor pairs, $bl=T_c|\{n \leftrightarrow m | n \in knn_p, m \in knn_q\}|$, to make sure the similarly level $sl(p,q)$ always greater than zero. The two best-matched local structure pairs $\{f_{sp} \leftrightarrow f_{tq}\}$ and $\{f_{su} \leftrightarrow f_{tv}\}$ are obtained by maximizing the similarity level. Experimental results in Fig. 3 to Fig. 6 confirm the improvement.

2.2.3 Global structure matching

There are two limitations in the local structure matching: first, two different feature points from the sensed and template images may have similar local structure. Second, two images from the same scene may have only a small number of well-matched local structures. We need to apply the global structure matching.

Assume that we obtain two best-matched local structure pairs, say *(p,q),*and *(u,v),* from the local structuring matching, either one of them can serve as a reliable correspondence of the two feature points' sets. We perform the global structure matching in two cues for consistence. The best-matched local structure pair (*p, q*) is sent to cue 1 as the corresponding reference to align two feature sets, while another best-matched local structure pair (*u, v*)is sent to cue 2 for the same purpose. In cue1, all feature points will be aligned as follows:

$$GS_i^s = \begin{pmatrix} r_{ip} \\ \theta_{ip} \\ \varphi_{ip} \end{pmatrix} = \begin{pmatrix} \sqrt{(x_i - x_p)^2 + (y_i - y_p)^2} \\ d\phi(\tan^{-1}(\frac{y_i - y_p}{x_i - x_p}), \varphi_p) \\ d\phi(\varphi_i, \varphi_p) \end{pmatrix}, \quad GS_i^t = \begin{pmatrix} r_{iq} \\ \theta_{iq} \\ \varphi_{iq} \end{pmatrix} = \begin{pmatrix} \sqrt{(x_i - x_q)^2 + (y_i - y_q)^2} \\ d\phi(\tan^{-1}(\frac{y_i - y_q}{x_i - x_q}), \varphi_q) \\ d\phi(\varphi_i, \varphi_q) \end{pmatrix}, \quad \textbf{(9)}$$

where GS_i^s and GS_i^t represent the aligned global structure of feature points *i* and *j* in sensed image *s* and template image *t* according to the corresponding reference *p* and *q*, respectively.

Then we define the matching level *ml*(*i,j*) for feature point *i* of the sensed image *s* and feature point *j* from the template image *t* by:

$$ml(i,j) = \begin{cases} 0.5 + 0.5 * w \mid GS_i^s - GS_j^t \mid, & \text{if } \mid GS_i^s - GS_j^t \mid < Bg \\ 0, & \text{otherwise} \end{cases} \quad \textbf{(10)}$$

where *w* is a weight vector and B_g is a 3-D bounding box in the feature space to tolerate the image deformation. We empirically choose $B_g = (10, \pi/4, \pi/4)$.

Thus for an arbitrary feature point *a* in the feature sets $\boldsymbol{F}_s$, we find a feature points *b* in the feature sets $\boldsymbol{F}_t$ such that $ml(a,b)=max_j\ (ml(i,j))$. While for this feature point *b*, we search for a feature point *c* in the feature sets $\mathbf{F}_s$ such that $ml(b,c)=max_i\ (ml(i,j))$.

The feature point *a* and the feature point *b* will be recognized as a matching pair if and only if the feature points *c* and *a* are the same point. A matching pair set *MP1* containing all correspondences is generated as the output of cue 1.

In cue 2, we align the two feature sets with respect to another corresponding reference f_u and f_v, and then perform the same global matching as what we did in cue 1 to generate the matching pair set *MP2*. Only those pairs are found in both cues are considered as valid matching pairs. Finally, the matching pair set *MP*, which is the intersection *MP1* and *MP2,* is the result of the global structure matching.

2.2.4 Eliminating the low-quality matching pairs

We have obtained a number of matching pairs from the global structure matching. Now, we apply the validation step to eliminate those low-quality matching pairs by cross-validation. First of all, we compute the mapping parameters (say, *Map*) from the whole matching pair set. Then in each step, we exclude one pair (say, P_i) from the set of matching pairs and compute the mapping parameters (say, Map_i). If the displacement between Map_i and *Map* is beyond a threshold, the matching pair P_i is identified as a low-quality matching. Eliminating them we get the correct matching pair set *MP'*. The experimental results are shown in Fig. 7 and Fig. 8.

2.3 Transformation model estimation

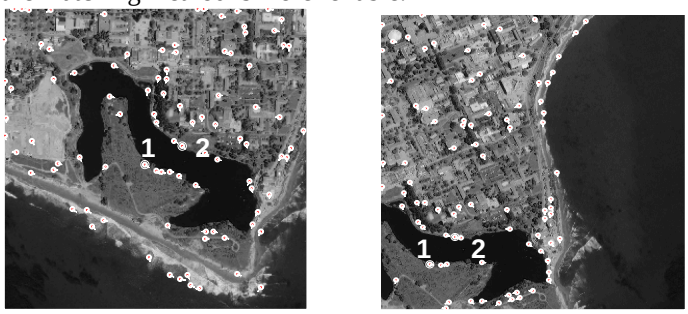
Assume that the matching pair set obtained is $\{u_i \leftrightarrow v_i\}_{i=1,2,\ldots,N}$. They should satisfy the relation $v_i = u_i A$, where A is the mapping function corresponding to the geometric transformation of the images. We compute it by least-square QR factorization.

3 Experimental study

A series of experiments are conducted. The majorities of our testing images are from [2], including optical, radar, multi-sensor, high-resolution and Landsat images. The testing platform is a Pentium 2.20GHz, 512MB RAM PC.

3.1 Results of local structure matching

To demonstrate the improvement of the local structure matching, we ran tests on the following four pairs of images with different types of image variations. The results are shown in Fig. 3 to Fig. 6. For every pair, the dots and arrows indicate the positions and orientations of the feature points, and the two best-matched local structure pairs computed are circled and numbered. The variations between the input and the template image and the number of feature points detected are listed in Table 1. From the results, we see that the best-match local structure pair computed by the improved local structure matching method is more reliable.



(a) template image (b) input image

Fig. 3. An example of local structure matching on images with geometry transformation.

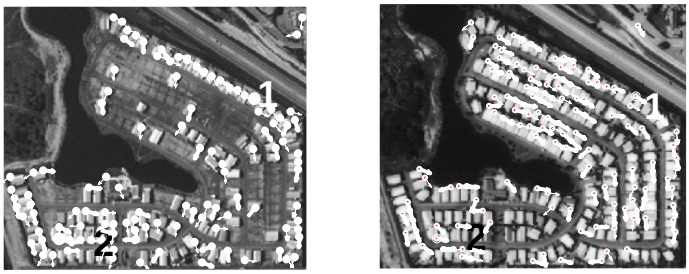


Fig. 4. An example of local structure matching on images with highly temporal changes.

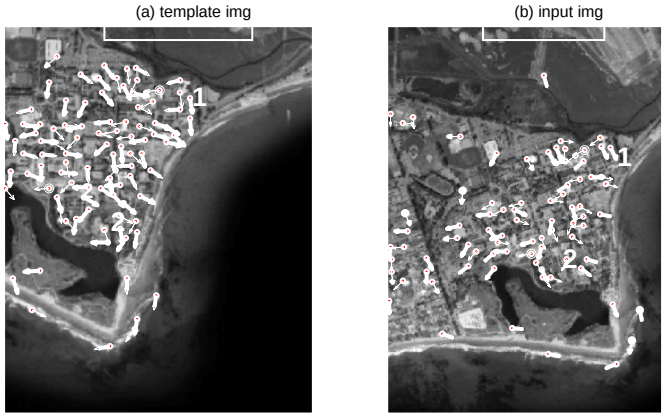


Fig. 5. An example of local structure matching on images with serious deformation.

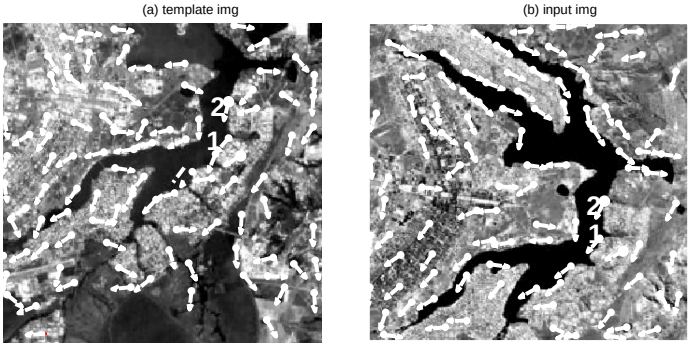


Fig. 6. An example of local structure matching on images from different sensors. In (a) SPOT band 3; (b) TM band 4.

To compare the performance of the two local structure matching methods (subsection 2.2.2), we present Table 1 of the experiments results on the four pair of images by both methods. The variation between the input and the template images are shown in the 2nd column. The number of feature points used is listed in the 3rd column. In the 4th and 5th column, method 1 is the direct local structure matching and method 2 is the complex local structure matching. For each method the time of computing the best-local structure pair is listed, where × means the corresponding method fails to compute the best-matched local structure pair.

Table 1. Comparisons of the two local structure matching methods.

Testing Images	Image variation type	#Feature points	Method 1	Method 2
Fig. 3	transformation	95 and 86	2.04s	20.34s
Fig. 4	temporal change	114 and139	×	57.80s
Fig. 5	distortion	96 and 81	×	5.25s
Fig. 6	different sensors	97 and 103	×	46.63s

From Table 1, we can see that the direct local structure matching method fails on images with significant scene changes, while the complex local structure matching method is applicable in those cases. However, the direct matching method is more

efficient of $O(kmn)$. The computation time of the complex matching method is $O(k^2mn)$, where k is the number of feature points in a neighborhood, m and n are the number of feature points in the input and template images.

3.2 Results of global structure matching

In this subsection, we show how the reliability of the feature points matching is improved by the global structure matching and cross validation. The testing image is pair of urban images from SPOT and TM (Fig. 7), where the two align references pairs are shown in Fig. 6. The final result of global structure matching is shown in Fig. 8.

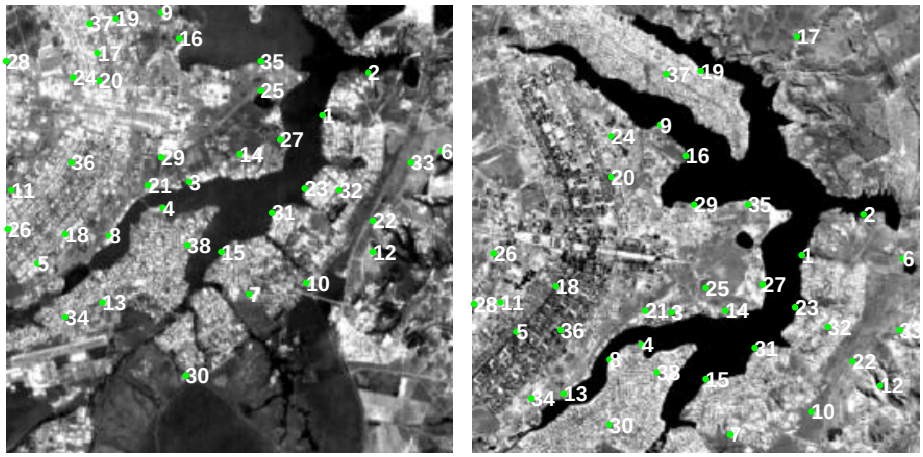


Fig. 7. The matching pairs detected from the global structure matching in cue 1.

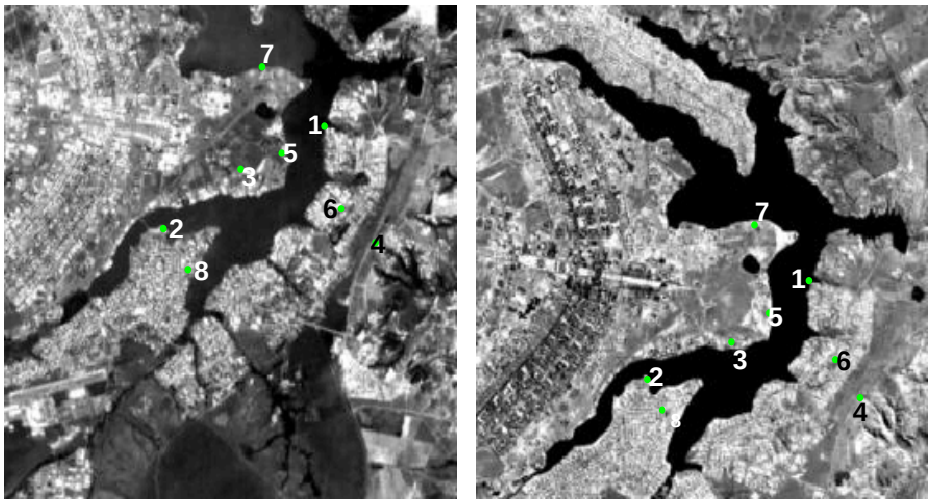


Fig. 8. The final matching pair set after cross-validation.

3.3 Using image mosaic for the registration results

We use image mosaic to show the registration results intuitively. Because of the page limit, only two examples of our test are shown in Fig. 9 and Fig. 10. The correctness of the registration results can be verified visually by checking the continuity of the common edges and regions in the mosaic images.

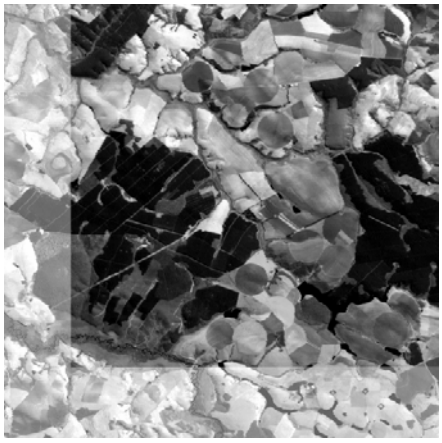


Fig. 9. Registration of Landsat images with four year difference and associated rotation.

Fig. 10. Registration of images with serious distortions.

We also compared the registration results generated by the UCSB automatic registration system [2]. In Table 2, $[s_1,t_{x1},t_{y1},\theta_1]$ are the registration parameters generated by our method and $[s_2,t_{x2},t_{y2},\theta_2]$ are the results by the UCSB system. In the last two column of Table 2, REMS is the root mean square error at the matching pairs. #MP indicates the number of matching pairs detected for each pair of images. It shows that our method performs better.

Table 2. The registration results on 8 pairs of images.

Test cases	Scale: s		Translation t_x		Translation t_y		Rotation: θ		REMS	#MP
	s_1	s_2	t_{x1}	t_{x2}	t_{y1}	t_{y2}	θ_1	θ_2		
1	1.002	1.002	715.1	714.9	-489.66	-490.67	-25.02	-24.98	1.607	211
2	1.012	0.996	87.07	75.06	9.83	9.57	-1.234	-1.098	4.192	8
3	1.042	0.997	21.49	22.35	-8.205	-8.937	-0.668	-0.168	1.498	6
4 (Fig 9)	0.994	0.991	87.88	87.65	-78.98	-79.30	0.125	0.193	1.081	19
5	1.020	0.997	-4.12	0.020	2.064	-0.625	0.562	0.291	11.751	13
6	0.991	0.991	33.57	34.24	-183.43	-186.24	0.984	1.032	9.428	17
7	0.998	0.997	1.44	1.84	-3.17	-0.91	-0.269	-0.047	2.185	8
8 (Fig 10)	0.997	1.004	144.90	144.86	75.33	74.22	-19.90	-20.20	1.611	21

4 Conclusion

Image registration is an important operation in multimedia system. We have presented a feature-based image registration method. Compared to the conventional feature-based image registration methods, our method is robust by guaranteeing the high reliable feature points to be selected and used in the registration process. We have successfully applied our method to images of different conditions.

5 Acknowledgment

We want to thank Dr. Tong San Koh of NTU for discussions, Dr. Wee Kheng Leow and Dr. Alan Cheng Holun of NUS for comments on Nan Peng's MSc thesis, and comments from MMM 2007 annoymouus reviewers. Nan Peng was supported by NUS scholarship for graduate study.

References

1. L. G. Brown, A survey of Image Registration Techniques, ACM Computing Surveys, vol.24, pp. 326-376, 1992.
2. Dmitry V. Fedorov, Leila M. G. Fonseca, Charles Kenney, and B.S. Manjunath, Automatic image registration and mosaicking system, http://nayana.ece.ucsb.edu/registration/, 2001-2004.
3. Intel Corporation. Open source computer vision library reference manual, December 2000. http://sourceforge.net/projects/opencvlibrary/.
4. X. D. Jiang and W. Y. Yau, Fingerprint Minutiae Matching Based on the Local and Global Structures. 15th International Conference on Pattern Recognition (ICPR'00), vol. 2, pp.1038-1041, 2000.
5. B. Zitova and J. Flusser, Image Registration Methods: a Survey, Image and Vision Computing, vol.21, pp. 977-1000, 2003.