Point pattern matching based on kernel partial least squares

Weidong Yan (延伟东)[1*], Zheng Tian (田 铮)[1,2], Lulu Pan (潘璐璐)[1], and Jinhuan Wen (温金环)[1]

[1]*School of Science, Northwestern Polytechnical University, Xi'an 710072, China*

[2]*State Key Laboratory of Remote Sensing Science, Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing 100101, China*

[*]*Corresponding author: weidongyan@qq.com*

Received July 5, 2010; accepted August 3, 2010; posted online January 1, 2011

Point pattern matching is an essential step in many image processing applications. This letter investigates the spectral approaches of point pattern matching, and presents a spectral feature matching algorithm based on kernel partial least squares (KPLS). Given the feature points of two images, we define position similarity matrices for the reference and sensed images, and extract the pattern vectors from the matrices using KPLS, which indicate the geometric distribution and the inner relationships of the feature points. Feature points matching are done using the bipartite graph matching method. Experiments conducted on both synthetic and real-world data demonstrate the robustness and invariance of the algorithm.

OCIS codes: 100.0100, 100.2000.
doi: 10.3788/COL201109.011001.

Point matching and correspondence problems arise in various application areas such as computer vision, pattern recognition, machine learning, and so on. In 1993, Cox *et al.* gave an overview of the different types of transformations and methods[1]. The advantage of point set representations of shapes over other forms such as curves and surfaces is that the point set representation is a universal representation of shapes regardless of the topologies of the shapes[2].

Given two or more images represented using points, we can match these point patterns to determine a transformation between coordinates of the point sets. Such transformations capture changes in the geometry characterized by the given points. Feature points are the simplest form of features, which are basically represented by the point locations. Feature points often serve as the basis upon which other more sophisticated representations (such as curves and surfaces) are built. They can also be regarded as the most fundamental of all features. Topological and geometrical relations between features contain important information to constrain the large space of possible mappings between features[3]. Feature points can be matched either by considering the radiometric properties of surrounding pixels or the geometric distribution of the whole set of feature points across the image[4]. Spectral graph theory is a term applied to a family of techniques that aim to characterize the global structural properties of graphs using eigenvalues and eigenvectors of similarity matrices[5,6]. The spectra can indicate the geometric distribution of feature points. Graph spectral methods have been extensively used for correspondences between feature point sets[7−9]. Scott *et al.* first used a Gaussian weighting function to build an inter-image similarity matrix between feature points in different images being matched, and performed singular value decomposition on the similarity matrix to get correspondences from the similarity matrix's singular values and vectors[7]. However, this method fails when the rotation or scaling between the images is too large. To overcome this problem, Shapiro *et al.* constructed intra-image similarity matrices for the individual point sets being matched with the aim of capturing the relational image structure[8]. The eigenvectors of the individual similarity matrices are used in the matching process. This method projects the individual point sets into an eigenspace and seeks matches by looking for the closest point correspondence. Wang *et al.* investigated the performance of the kernel principal component analysis (PCA) using a polynomial kernel function for solving the point correspondence problem[9]. Such approaches characterize the graphs by their dominant eigenvectors. However, these eigenvectors are computed independently for each graph, thus they often do not capture the co-salient structures of the graph. The kernel partial least squares (KPLS) approach helps extract representations from the two images containing relevant information needed for the matching of the particular pair of images.

This letter presents a method of spectral feature matching based on KPLS. Using KPLS components, the spectral features are constructed with information on the position similarity matrices, invariant to translation, scale, and rotation, and very suitable for feature-based matching.

In general, intrinsic point representations can be obtained via the position similarity matrix, specified by a symmetric affinity matrix $\mathbf{S} = \{s_{ij}\}$, where $s_{ij} \geq 0$ characterizes the similarity between points x_i and x_j. One may view the position similarity matrix $\mathbf{S}$ as a data vector whose n rows represent n-dimensional data points.

Although the position similarity matrices contain considerable shape information, one cannot compare such representations for two point sets directly without proper point mapping. The high dimensional representations may contain a great deal of redundancy, resulting in unnecessarily high computational costs[10]. These observations naturally lead us to consider transforming two point sets into some information-preserving subspaces. This can be accomplished through KPLS on the position similarity matrices.

Partial least squares (PLS) method, which was initially developed by Wold *et al.*[11], has been a tremendously successful method for data analysis in the chemometrics and chemical industries. The robustness of the generated model also makes the PLS approach a powerful tool for dimensionality reduction, discrimination, and classification technique; it is being applied to many other areas such as process monitoring and image processing[12−14]. In its general form, PLS creates components using the existing correlations between different sets of variance while keeping most of the variance of both sets. KPLS is a modification of PLS through the use of kernels. Following the theoretical and practical results reported in Ref. [14], KPLS seems to be preferred to the kernel PCA when a feature space dimensionality reduction with respect to data discrimination is employed.

We consider a general setting of the PLS algorithm to demonstrate the relation between two data sets. An N-dimensional vector of variables is denoted as $\mathbf{x} = (x_1, \cdots, x_N)$ in the first block of data, and similarly, $\mathbf{y} = (y_1, \cdots, y_N)$ denotes a vector of variables from the second set. Observing the n data samples from each block of variables, PLS decomposes $\mathbf{X} = (x_{ij})_{n\times N}$, $\mathbf{Y} = (y_{ij})_{n\times N}$ into the form

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{F},$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^{\mathrm{T}} + \mathbf{G},$$

where $\mathbf{T}$ and $\mathbf{U}$ are $(n \times r)$ matrices of the extracted r components, and the $(N \times r)$ matrices $\mathbf{P}$ and $\mathbf{Q}$ represent the matrices of the projections. The $n \times N$ matrices $\mathbf{F}$ and $\mathbf{G}$ are the matrices of the residuals. The PLS method, which in its classical form is based on the nonlinear iterative partial least squares (NIPALS) algorithm, finds the projection axes $\mathbf{w}$ and $\mathbf{c}$ such that[11]

$$\begin{array}{ll} \max & \mathbf{S} = \mathbf{t}^{\mathrm{T}}u = (\mathbf{X}\mathbf{w})^{\mathrm{T}}(\mathbf{Y}\mathbf{c}) = \mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{Y}\mathbf{c} \\ s.t. & \left\{ \begin{array}{l} \mathbf{w}^{\mathrm{T}}\mathbf{w} = \|\mathbf{w}\|^2 = 1 \\ \mathbf{c}^{\mathrm{T}}\mathbf{c} = \|\mathbf{c}\|^2 = 1 \end{array} \right. \end{array} . \qquad (1)$$

The solution to this optimization problem is given by the following eigenvalue problem[15]:

$$\mathbf{X}^{\mathrm{T}}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\mathbf{X}\mathbf{w} = \lambda\mathbf{w},$$

where λ is the eigenvalue associated with $\mathbf{w}$. The component of $\mathbf{X}$ is then given as $\mathbf{t} = \mathbf{X}\mathbf{w}$.

Similarly, the extraction of the component of $\mathbf{Y}$ is given as

$$\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\mathbf{t} = \lambda\mathbf{t}, \qquad (2)$$

$$\mathbf{u} = \mathbf{Y}\mathbf{Y}^{\mathrm{T}}\mathbf{t}. \qquad (3)$$

Like other kernel-based algorithms, KPLS is based on mapping the original data into feature space, in which a linear regression function is constructed. We consider a nonlinear transformation of x into a feature space $\mathbf{F}$. In this case, the projection axes $\mathbf{w}$ and $\mathbf{c}$ cannot be computed. Using the straightforward connection between a reproducing kernel hilbert space and $\mathbf{F}$, Rosipal *et al.* extended the linear PLS model into its nonlinear kernel form[16]. This extension effectively represents the construction of a linear PLS model in $\mathbf{F}$.

The feature space $\mathbf{F}$ could have an arbitrarily large, possibly infinite dimensionality related to the data space by a possibly nonlinear map $\phi : x \to \phi(x)$. $\mathbf{\Phi}$ denotes the matrix of the mapped data space $\phi(x)$ into a feature space $\mathbf{F}$. Using the kernel method, we can obtain $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}$, where $\mathbf{K}_{ij} = \mathbf{K}(x_i, x_j)$ is the Gram matrix.

Similarly, we consider a mapping of the second set of variables y into a feature space $\mathbf{F}_1$, and denote it as $\mathbf{\Psi}$, the matrix of the mapped data space $\psi(y)$ into a feature space $\mathbf{F}_1$. Analogous to $\mathbf{K}$, we define the kernel Gram matrix as $\mathbf{K}_1 = \mathbf{\Psi}\mathbf{\Psi}^{\mathrm{T}}$, given by the kernel function $\mathbf{K}_1(\cdot, \cdot)$. We can use $\mathbf{K} = \langle \mathbf{\Phi}, \mathbf{\Phi} \rangle$ and $\mathbf{K}_1 = \langle \mathbf{\Psi}, \mathbf{\Psi} \rangle$ instead of $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ and $\mathbf{Y}\mathbf{Y}^{\mathrm{T}}$ because $\mathbf{X}\mathbf{X}^{\mathrm{T}} = \langle \mathbf{X}, \mathbf{X} \rangle$ and $\mathbf{Y}\mathbf{Y}^{\mathrm{T}} = \langle \mathbf{Y}, \mathbf{Y} \rangle$ are inner product forms. Using this notation, the estimates of $\mathbf{t}$ and $\mathbf{u}$ can be reformulated into its nonlinear kernel variant (KPLS) from Eqs. (2) and (3):

$$\mathbf{K}\mathbf{K}_1\mathbf{t} = \lambda\mathbf{t},$$

$$\mathbf{u} = \mathbf{K}_1 t.$$

When the data in the feature space do not contain zeros, the mean $\frac{1}{n}\sum\limits_i \mathbf{\Phi}(x_i)$ is subtracted from all points. This leads to a slightly different expression

$$\mathbf{K} = \big(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}\big)\mathbf{K}\big(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}\big),$$

where $\mathbf{1}_n = (1, \cdots, 1)^{\mathrm{T}}$.

Similarly, $\mathbf{K}_1 = (\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}})\mathbf{K}_1(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}})$.

We select feature point sets from the reference image and the sensed image, and denote them as $\mathbf{X} = (x_1, x_2, \cdots, x_n)$ and $\mathbf{Y} = (y_1, y_2, \cdots, y_n)$, respectively. Our objective is to establish a one-to-one point correspondence between the two data sets.

Using the feature point sets, we construct position similarity matrices by the Gaussian kernel function, $(\mathbf{S}_x)_{n\times n}$ and $(\mathbf{S}_y)_{n\times n}$:

$$(\mathbf{S}_x)_{ij} = \exp\big[-\frac{d(x_i, x_j)^2}{\sigma_x^2}\big],$$

$$\text{and } (\mathbf{S}_y)_{ij} = \exp\big[-\frac{d(y_i, y_j)^2}{\sigma_y^2}\big],$$

where $d(x_i, x_j)$ is the Euclidean distance between the feature points x_i and x_j, and σ_x, σ_y are the adjustable parameters. Every feature point corresponds to an $n$dimensional feature vector, $x_i \to (\mathbf{S}_x)_i$, $y_j \to (\mathbf{S}_y)_j$.

Theorem 1. Under criterion (1), the number of components is $\mathrm{rank}(\mathbf{S}_x^{\mathrm{T}}\mathbf{S}_y\mathbf{S}_y^{\mathrm{T}}\mathbf{S}_x)$ (the number of non-zero eigenvalues of matrix $\mathbf{S}_x^{\mathrm{T}}\mathbf{S}_y\mathbf{S}_y^{\mathrm{T}}\mathbf{S}_x$) pairs at most. $r(\leq \mathrm{rank}(\mathbf{S}_x^{\mathrm{T}}\mathbf{S}_y\mathbf{S}_y^{\mathrm{T}}\mathbf{S}_x))$ pairs of components are composed of vectors selected from the eigenvectors corresponding to the first r maximum eigenvalues of the eigenequations

$$\mathbf{S}_x\mathbf{S}_x^{\mathrm{T}}\mathbf{S}_y\mathbf{S}_y^{\mathrm{T}}\mathbf{T} = \lambda^2\mathbf{T}, \qquad (4)$$

$$\mathbf{S}_y\mathbf{S}_y^{\mathrm{T}}\mathbf{S}_x\mathbf{S}_x^{\mathrm{T}}\mathbf{U} = \lambda^2\mathbf{U}, \qquad (5)$$

where $\mathbf{T} = [t_1, \cdots, t_r]$, $\mathbf{U} = [u_1, \cdots, u_r]$.

See Ref. [17] for detailed proof.

Similarly, let $\mathbf{K} = \mathbf{S}_x\mathbf{S}_x^{\mathrm{T}}$, $\mathbf{K}_1 = \mathbf{S}_y\mathbf{S}_y^{\mathrm{T}}$. Thus, we can derive Theorem 2 from Theorem 1 easily.

Theorem 2. Under criterion (1), the number of components is $\mathrm{rank}(\mathbf{S}_x^{\mathrm{T}}\mathbf{S}_y\mathbf{S}_y^{\mathrm{T}}\mathbf{S}_x)$ (the number of non-zero eigenvalues of matrix $\mathbf{S}_x^{\mathrm{T}}\mathbf{S}_y\mathbf{S}_y^{\mathrm{T}}\mathbf{S}_x$) pairs at most. $r(\leq \mathrm{rank}(\mathbf{S}_x^{\mathrm{T}}\mathbf{S}_y\mathbf{S}_y^{\mathrm{T}}\mathbf{S}_x))$ pairs of components are composed of vectors selected from the eigenvectors corresponding to the first $r$maximum eigenvalues of eigenequations

$$\mathbf{K}\mathbf{K}_1\mathbf{T} = \lambda\mathbf{T},$$

$$\mathbf{K}_1\mathbf{K}\mathbf{U} = \lambda\mathbf{U},$$

where $\mathbf{T} = [t_1, \cdots, t_r]$, $\mathbf{U} = [u_1, \cdots, u_r]$.

The feature point x_i in the reference image corresponds to the KPLS pattern vector $(\mathbf{T})_i = [t_{i1}, t_{i2}, \cdots, t_{ir}]$, which are the projections in the r-dimensional eigenspace spanned by $\mathbf{T}$. The feature point y_j in the sensed images corresponds to the KPLS pattern vector $(\mathbf{U})_j = [u_{j1}, u_{j2}, \cdots, u_{jr}]$, which is the projections in the r-dimensional eigenspace spanned by $\mathbf{U}$. Thus, if the arbitrary numbering of two feature points in an image is changed, their KPLS pattern vectors simply change positions in $\mathbf{T}$(or $\mathbf{U}$), and the matching of the feature points can be converted to the matching of the KPLS pattern vectors. The Euclidean distance is invariant to the similarity transformation[10]. Hence, the KPLS pattern vectors have invariance.

In the last decade, numerous techniques have been proposed to tackle the feature point matching problem, such as iterative closest point (ICP)[18], graph matching[19−21], and cluster approach[22]. Among these methods, the ideas of the graph matching algorithm are very attractive. In graph matching, feature points are modeled as graphs, and feature matching amounts to finding a correspondence between the nodes of different graphs. In this letter, bipartite matching method[23] is applied in feature matching. We extract feature points from the reference and sensed images. The KPLS pattern vectors ($\mathbf{T}$ and $\mathbf{U}$) are then computed for the feature point sets as vertices for a bipartite graph. We apply bipartite graph matching algorithms to this graph to obtain the set of edges in the optimal matching, which represents the correspondence.

To make the algorithm more robust, the mismatching feature points can be eliminated by the local continuity constraint (the continuity constraint requires that neighboring points in the reference match neighboring points in the sensed). This constraint is reasonable for most images. We can examine the correspondences of the neighbors of the matched interest points to eliminate mismatching.

Now we investigate the performance of the method of point pattern matching based on KPLS. In experiments, the robustness of the KPLS approach described above with random point sets is shown, the invariance of the KPLS pattern vectors on synthetic images is verified, the matching performances of the KPLS approach described above and the method of Ref. [9], are compared, and the remote sensing image. In addition, we use this algorithm to solve the non-rigid mapping.

We investigated the effect of the controlled affine skew of the point sets. The reference point set was randomly generated. Then, the reference point set was transformed by parameters (translation_x=0.1, translation_y=0.1, rotation_angle=30°, scaling_factor=1.2) to obtain the sensed data set. Figure 1 shows the matching results. We then focused on the performance of the algorithms when the data were under affine transformations and contained uncertainties such as outliers and noise. For this purpose, we added noise to the sensed data. The reference data set is the same as that used above, and five outliers were added to the sensed data set. Figure 2 shows the matching results. It can be seen that the KPLS algorithm presents a strong ability to eliminate incorrect matches.

To provide more quantitative evaluations, we also tested the algorithm on synthetic images. Here, we have matched images from a gesture of a hand. The feature

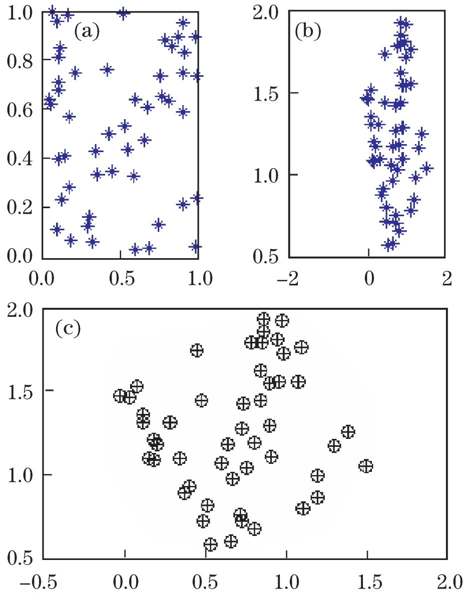

Fig. 1. (a) Reference data set and (b) sensed data set without outliers, and (c) the match between the two data sets found with our implementation. The circles in (c) are the reference data and the crosses are the sensed data.

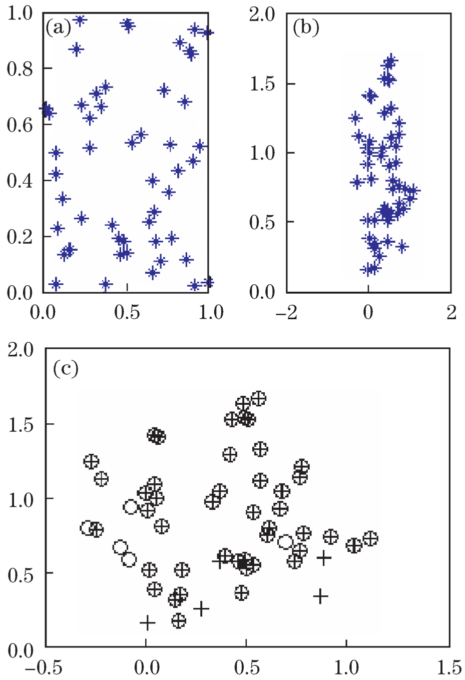

Fig. 2. (a) Reference data set and (b) sensed data set with outliers, and (c) the match between the two data sets found with our implementation.

points in these images are points of maximum curvature on the outline of the hand. Figure 3 shows the final configuration of correspondence matches obtained using our method. From these experiments, we can see that the KPLS pattern vectors are invariant to rotation and scaling.

We tested our algorithm on the Carnegie Mellon University (CMU) house sequence, and compared the performance of our method with that of the method of Ref. [9]. The images used in our study correspond to different camera viewing directions. The corner points in each image were detected by the Harris detector. The first frame was tested against the remaining frames. Figure 4 shows the correspondences when we matched the first frame to the fourth and the sixth frames. The experimental results are summarized in Table 1. From these results, our method clearly performs better than Wang and Hancock's method.

To test our algorithm on a real world situation, we applied our method to a two-dimensional (2D) image registration with an affine model. We took two images from

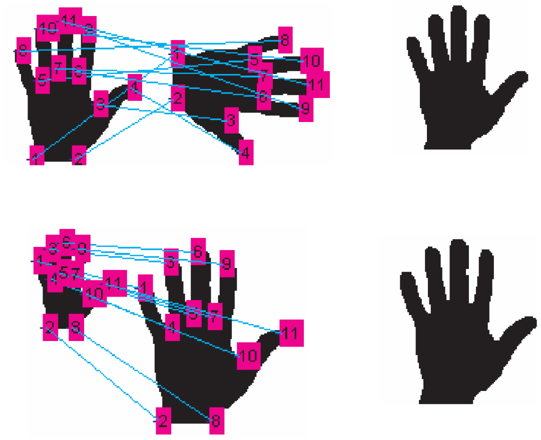

Fig. 3. (Point matching result on the hand images. Top row: the correspondences between the hand-rotation (left) and the registration result (right). Bottom row: the correspondences between the hand-scaling (left) and the registration result (right).

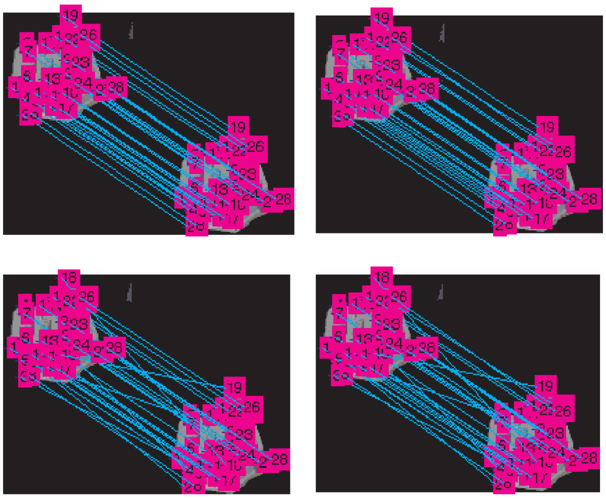

Fig. 4. Point matching result on the CMU house sequence. Top row: our approach. Bottom row: the kernel PCA method.

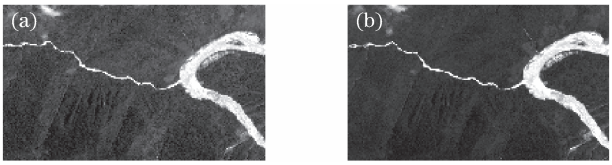

Fig. 5. (a) Band-1 and (b) band-2 images.

Table 1. Summary of the Experimental Results on the CMU House Sequence

Image	1−2	1−3	1−4	1−5	1−6
Correctly Matched (Our Approach)	28	22	28	28	26
Rate of Matched (Our Approach) (%)	100	79	100	100	93
Correctly Matched (Method of Ref. [9])	24	20	22	22	20
Rate of Matched (Method of Ref. [9]) (%)	86	71	79	79	71

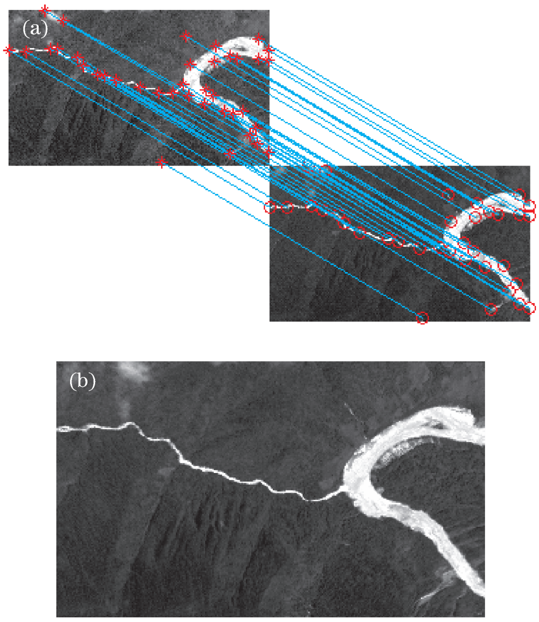

Fig. 6. (a) Correspondences of feature points between the images and (b) registration result computed with the proposed approach.

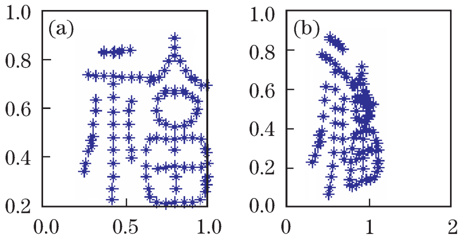

Fig. 7. (a) Reference image and (b) sensed image.

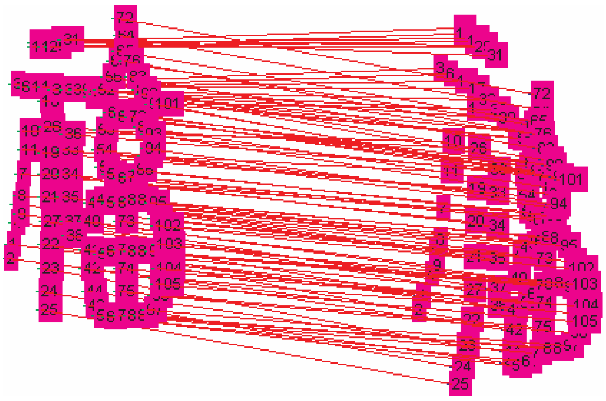

Fig. 8. Correspondences of points between the images.

the same area. The feature points in each image were detected by a Harris detector. We then tried to match them using our implementation. Figure 5 shows the two multi-spectral images (band-1 and band-2). Figure 6 shows our registration result.

We tested our algorithm on non-rigid transformations. The reference image that we chose comes from a Chinese character (blessing), which is a rather complex pattern. Figure 7 shows the character images and Fig. 8 shows the results of the matching.

In conclusion, we investigate the spectral approaches to the problem of point pattern matching. First, we show how KPLS can be effectively used for solving the point pattern matching problem. We use the KPLS pattern vectors of feature point sets from both reference and sensed images to establish their correspondences. The KPLS pattern vectors indicate the geometric distribution of the feature points, and are invariant to translation, scale, and rotation. We then use robust methods for the points correspondence by the continuity constraint. Results show that the method can deal with non-rigid transformation. Further work has to be done in order to obtain more robust results using more matching feature points to get better mapping approximations and to work with feature point sets of different sizes.

This work was supported by the Northwestern Polytechnical University Doctoral Dissertation Innovation Foundation (No. CX200819), the National Natural Science Foundation of China (Nos. 10926197 and 60972150), and the Science and Technology Innovation Foundation of Northwestern Polytechnical University (No. 2007KJ01033).

References

1. G. S. Cox and G. De Jager, in *Proceedings of South African Symposium on Communications and Signal Processing* 243 (1993).
2. H. Guo, A. Rangarajan, and S. Joshi, “Diffeomorphic point matching” in *Handbook of Mathematical Models in Computer Vision* (Springer, Heidelberg, 2005) pp.205−220.
3. J. You and P. Bhattacharya, IEEE Trans. Image Process. **9,** 1547 (2000).
4. P. Dare and I. Dowman, ISPRS J. Photogramm. Remote Sens. **56,** 13 (2001).
5. F. R. K. Chung, *Spectral Graph Theory* (American Mathematical Society, Providance, 1997).
6. M. Carcassoni and E. R. Hancock, Pattern Recognition **36,** 193 (2003).
7. G. L. Scott and H. C. Longuet-Higgins, Proc. R. Soc. Lond. B **244,** 21 (1991).
8. L. S. Shapiro and J. M. Brady, Image and Vision Computing **10,** 283 (1992).
9. H. Wang and E. R. Hancock, LNCS **3138,** 361 (2004).
10. V. Jain and H. Zhang, in *Proceedings of 2006 IEEE International Conference on Shape Modeling and Applications* 118 (2006).
11. S. Wold, A. Ruhe, H. Wold, and W. J. Dunn III, SIAM J. Sci. Stat. Comput. **5,** 735 (1984).
12. D. V. Nguyen and D. M. Rocke, Bioinformatics **18,** 39 (2002).
13. Q.-S. Sun, Z. Jin, P.-A. Heng, and D.-S. Xia, LNCS **3686,** 268 (2005).
14. M. Barker and W. Rayens, J. Chemometrics **17,** 166 (2003).
15. A. Hoskuldsson, J. Chemometrics **2,** 211 (1988).
16. R. Rosipal and L. J. Trejo, Machine Learning Res. **2,** 97 (2002).
17. W. Yan, Z. Tian, L. Pan, and M. Ding, Chin. Opt. Lett. **7,** 201 (2009).
18. P. J. Besl and N. D. McKay, IEEE Trans. Pattern Anal. Mach. Intell. **14,** 239 (1992).
19. T. Jebara and V. Shchogolev, LNAI **4212,** 679 (2006).
20. T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola, IEEE Trans. Pattern Anal. Mach. Intell. **31,** 1048 (2009).
21. C. Leng, Z. Tian, J. Li, and M. Ding, Chin. Opt. Lett. **7,** 996 (2009).
22. S.-H. Chang, F.-H. Cheng, W.-H. Hsu, and G.-Z. Wu, Pattern Recognition **30,** 311 (1997).
23. Y. Wang, F. Makedon, and J. Ford, in *Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* **2,** 2972 (2004).