

Correspondence matching using kernel principal components analysis and label consistency constraints

Hong Fang Wang*, Edwin R. Hancock

Department of Computer Science, University of York, York YO10 5DD, UK

Received 8 February 2005; received in revised form 26 May 2005; accepted 26 May 2005

Abstract

This paper investigates spectral approaches to the problem of point pattern matching. We make two contributions. First, we consider rigid point-set alignment. Here we show how kernel principal components analysis (kernel PCA) can be effectively used for solving the rigid point correspondence matching problem when the point-sets are subject to outliers and random position jitter. Specifically, we show how the point-proximity matrix can be kernelised, and spectral correspondence matching transformed into one of kernel PCA. Second, we turn our attention to the matching of articulated point-sets. Here we show label consistency constraints can be incorporated into definition of the point proximity matrix. The new methods are compared to those of Shapiro and Brady and Scott and Longuet-Higgins, together with multidimensional scaling. We provide experiments on both synthetic data and real world data.

© 2005 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.

Keywords: Non-rigid motion; Correspondence matching; Graph spectral methods; Kernel PCA; Constraints

1. Introduction

The problem of point pattern matching is to find one-to-one correspondences between two given data-sets and serves as an important part in many computer vision tasks. Graph spectral methods [1] have been used extensively for locating correspondences between feature point-sets, e.g. Refs. [2,3,23,25,27]. In Ref. [2], Scott and Longuet-Higgins first use a Gaussian weighting function to build an inter-image proximity matrix between feature points in different images being matched and then perform singular value decomposition on the resulting matrix in order to locate correspondences from the singular values and vectors of the proximity matrix. This method fails when rotation or scaling between the images is too large. To overcome this problem, Pilu [4] incorporates a *feature similarity measure* into the algorithm. This is implemented by including

a neighbourhood correlation measure into the proximity matrix (the “correlation-weighted matrix”). The correspondences are then located using singular value decomposition operations. Shapiro and Brady [3] also improve the Scott and Longuet-Higgins method by constructing intra-image proximity matrices for the individual point-sets being matched with an aim to capturing relational image structure. In the latter approach, the eigenvectors of the individual proximity matrices are used as the columns of a modal matrix. Correspondences are located by comparing the rows of the modal matrices for the point-sets under match. This method can be viewed as projecting or embedding the individual point-sets into an eigenspace, and seeking matches by looking for closest embedded point correspondences. Carcassoni and Hancock have attempted to improve the robustness of the Shapiro and Brady method to point-jitter by using robust error kernels instead of the Gaussian [5]. They have also overcome problems due to differences in the structure of the point-sets by using spectral clusters [6]. Multidimensional scaling has also been used to solve this problem by performing Procrustes alignment in the eigenspace [7]. However,

* Corresponding author. Tel.: +44 1904 43 2795; fax: +44 1904 432767.

E-mail addresses: hongfang@cs.york.ac.uk (H.F. Wang),
erh@cs.york.ac.uk (E.R. Hancock)

URL: <http://www-users.cs.york.ac.uk/erh/>.

these two latter approaches involve iterative computation which may be more time-consuming. This observation leads us to seek point matching algorithms that are both robust to noise and structural error, and that are computationally more efficient.

The location of correspondences between feature points belonging to non-rigid objects is not only a more challenging but also a potentially more important task. Many existing approaches rely on such correspondence information. For example, the point distribution model (PDM) of Cootes and Taylor [8] requires reliable one-to-one feature point correspondences over a sequence of examples for the purposes of learning the modes of shape variation. The factorisation method of Tomasi and Kanade [9] also requires accurate correspondence information to separate motion and shape. Without an accurate means of locating the feature correspondences, the recovered model will be inaccurate. For instance, in point distribution models then the covariance matrix will represent the distribution of correspondence errors rather than the modes of shape variation. In the case of factorisation [9], then there will be errors in both the estimated motion and the recovered shape. In the literature, many attempts have been described to recover accurate correspondences for non-rigid motion. For example, in Ref. [10] the softassign method is used to compute correspondences in a manner that is robust to outliers.

Of course there is a wealth of information that can be exploited to improve and refine the point-correspondence process. If absolute position is used, then the detailed transformation between point-sets must be recovered. This is of course straightforward if the point-sets are known to undergo a rigid transformation, for example affine or perspective, or if there is a well-defined non-rigid transformation that can be applied, for example a spline-warp or a diffeomorphism. One way of avoiding the need to know the transformation geometry is to characterise the point-sets using information concerning their relational arrangement. Examples include proximity graphs [11] and proximity matrices [3,2] for rigid feature points. Since in non-rigid motion, structural deformation is one of the motion properties, additional information must be provided if the graph spectral approach is to be used. An interesting source of information that can be used, but has received relatively little attention, is that provided by label consistency constraints. In many types of image, the points can be assigned semantic labels to distinguish their identity. An example would be to assign labels to distinguish the limb types of articulated objects. Using this information the consistency of pairwise relations can be tested against a scene constraint model. Hence, correspondences which are inconsistent with the model can be rejected.

In this paper, we aim to develop a point pattern matching method which allows label consistency constraints to be incorporated into the point correspondence process. Our idea is to use a weighted adjacency matrix to represent each data-set. Here the weight reflects the label consistency information in addition to the common pairwise proximity re-

lationship. We use the label consistency weights to improve the matching process. The idea is that by guiding the correspondences of the feature points using label consistency constraints that apply to the rigid components, more effective matching results can be obtained. To do this we draw on ideas from probabilistic relaxation labelling [12,13,14]. We characterise each point by augmenting the positional information with a vector of label probabilities. In addition, the arrangement of the points is represented using a Gaussian point proximity matrix. Our first contribution is to show how the point-proximity matrix can be incorporated into the definition of the support function for relaxation labelling. In this way when the label probabilities are updated, then the strength of the proximity relations is brought to bear on the computation of label support. Our second contribution is to show how the label probabilities can be used to refine the point correspondence process. Here we use a kernelised version of the Shapiro and Brady algorithm [3]. We use the label probabilities to refine the kernel matrix used to locate point correspondences. The matching process is realised in an iterative fashion where there are interleaved steps for label probability update and for point correspondence matching.

The idea underpinning these spectral methods is to embed point-sets into a common eigenspace, and to find correspondences by performing alignment in this space. The key in this idea is that of finding the appropriate function which captures the essential properties of the given data-set which should also be robust under uncertainties such as outliers, random position jitter, occlusions, etc., and identifying the common eigenspace. Also the captured properties should be common in both data-sets. The problem of how to select the best kernel function is a topic that has recently attracted considerable interest in kernel learning theory. The development of kernel PCA [15] provides us with a theoretically sound way of improving the existing spectral point pattern matching algorithms since it shares many features in common with spectral graph theory.

Our aim in this paper is to investigate the performance of kernel PCA for solving the point correspondence problem. We first provide a robust one-to-one point pattern matching algorithm when there is only one label in each image. That is, the points are undergoing rigid motion. We show that without the need for iteration, the algorithm performs efficiently and effectively for locating correspondences. We extend the method to feature points from objects that undergo articulated motion. Label consistency information for the feature points is incorporated into the procedure. We focus in detail on Gaussian and polynomial kernels since these are invariant to similarity transformations. We also compare their performance with a number of previous approaches to point pattern matching. A common weakness with existing spectral correspondence methods is that they are particularly sensitive to structural variations in the point-sets under study. We demonstrate that with an appropriate choice of kernel function, the method delivers encourag-

ing performance. The results are less sensitive to the problems that limit the performance of previous graph spectral methods.

2. Spectral point pattern matching

The problem of point pattern matching can be described as given two feature point-sets \mathbf{y} and \mathbf{x} extracted from two different images, establish one-to-one point correspondences between the two data-sets. Ideally, outliers can be removed from the data-sets during matching. In this paper, the objects in which we are interested undergo articulated motion between image frames. That is, the objects under investigation are composed of rigid components. Each rigid component has its own rigid motion, but the overall motion is non-rigid. The feature points in each data-set in our work are given in the form of $\mathbf{y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ and $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ for the model point-set and data point-set, respectively. Each point $\mathbf{x}_i, i = 1, \dots, m$ is represented by their image co-ordinates; that is, $\mathbf{x}_i = (x_{i1}, x_{i2})$. The \mathbf{y}_i 's are also represented by their respective image co-ordinates. Furthermore, they also include each point label information which specifies to which rigid component a feature point is likely to belong. Thus $\tilde{\mathbf{y}}_j = (\mathbf{y}_j, \mathbf{p}_j)$ where \mathbf{p}_j is the label probability vector for point \mathbf{y}_j , and $\mathbf{y}_j = (y_{j1}, y_{j2})$. Assume there are L labels in each feature point-set, which represent the L rigid components possibly existing in each image frame, and that an image point \mathbf{x}_i can be assigned a label $\theta_i \in \Omega$, where $\Omega = \{\omega_1, \dots, \omega_L\}$. Denote by $P(\theta_i = \lambda)$ the probability that node \mathbf{x}_i is labelled as λ with $\lambda \in \Omega$. Then the vector $\mathbf{p}_i = (P(\theta_i = \omega_1), \dots, P(\theta_i = \omega_L))^T$ represents the probability of assigning each of the possible labels to the point, with $0 \leq P(\theta_i = \lambda) \leq 1$, and $\sum_{\lambda=1}^L P(\theta_i = \lambda) = 1$. The label probability matrix P has the probability vectors as columns, i.e., $P = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N)^T$, where N is the size of the corresponding data-set. It represents the label probability distribution over the entire point-set. Our ultimate aim is to locate correspondences between the two point-sets on the basis of the above-mentioned information and spectral graph theory.

Graph spectral methods for point pattern matching solve the point correspondence problem by first building a graph representation for each data-set. Each graph node corresponds to an image feature point. Each edge between nodes corresponds to a spatial relationship between the two feature points. After graph construction, we represent each point-set by a matrix and find feature correspondences using matrix eigendecompositions. These methods aim to embed the dissimilarity (or similarity) properties of the original data into a common space in which correspondence matching can be performed. As mentioned in the previous section, for the case of matching feature points undergoing rigid motion, the two essential ingredients are the dissimilarity function and the embedding procedure. Usually the dissimilarity properties are regarded as weights of the edges, and are expressed

in the form of a proximity matrix W . The elements w_{ij} of the matrix W represent the dissimilarity relationship between feature points \mathbf{x}_i and \mathbf{x}_j . One example is to follow Refs. [3,2], and to use the Gaussian function:

$$W_{ij} = e^{-d_{ij}^2/\sigma}, \quad (1)$$

where d_{ij} is the Euclidean distance between the two feature points \mathbf{x}_i and \mathbf{x}_j , and σ is a constant parameter. Another choice is to make a binary assignment of weights using the adjacency matrix. Accordingly $W_{ij} = 1$ if $(i, j) \in E$, i.e. the two points are connected by an edge in the graph, and $W_{ij} = 0$ otherwise.

In this paper we are interested in the weighted adjacency matrix. We propose to use the label information of each feature point to define the neighbourhoodship of it with other points when building this matrix representation.

Our aim in constructing the matrix representations for the point-sets is to provide a basis for the correspondence process. As we are dealing with objects subject to transformations such as translation, rotation, scaling and reflection, it is desirable for the dissimilarity function to be invariant under these transformations. It is known from geometry that the Euclidean distance is invariant to the similarity transformation. Hence, the dissimilarity functions underpinning many existing methods are related to the Euclidean distance between feature points (see, for example, Refs. [3,2]). Another example of the invariance of the similarity transformation is the directional properties of the feature points. This property can also be considered as a good candidate for constructing a suitable similarity function for spectral point pattern matching.

When viewed from the perspective of kernel principal components analysis (kernel PCA [15]), applying a dissimilarity or similarity function to the original data set is equivalent to the process of using a kernel function to map the data into a higher, possibly infinite, dimensional space. Moreover, this mapping interpolates the data in the new space according to their transformation invariant properties. From this perspective, kernel PCA appears to provide us with a sound theoretical basis for spectral pattern matching. In the next section we review the kernel PCA method.

3. The kernel method

Kernel PCA [15] can be regarded as a generalization of PCA from a linear to a non-linear transformation space. In the literature, it has been shown to provide a better way of recovering the underlying principal components of the given data, e.g., the de-noising application in Ref. [16].

Conventional principal components analysis (PCA) provides an orthogonal transformation of the data from a high dimensional space to a low dimensional one which maximally preserves the variance of the original data. This is

done by computing the eigenvalues and eigenvectors of the covariance matrix

$$C = \frac{1}{m-1} \sum_{i=1}^m (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

of a given data-set $X = \{\mathbf{x}_i\}_{i=1}^m$, and then using the first k normalized eigenvectors (where $k \leq m$ and the eigenvalues are sorted in descending magnitude order) of the covariance matrix as the principal projection axes for the training data. Since the method minimizes the residual covariance of the data points projected into the common eigen-subspace, it thus gives an optimum representation of the original data in the chosen projection space.

The main difference between kernel PCA and conventional PCA is that kernel PCA first uses a function $\mathcal{F}: \mathbf{x} \mapsto \Phi(\mathbf{x})$ to map the data from the low dimensional space into a new feature space \mathcal{F} of higher, possibly infinite, dimension. Conventional PCA is then performed on the transformed data matrix to obtain the data projection. This gives kernel PCA the property of extracting non-linear features from the data-set and makes it a powerful tool in many pattern analysis applications.

However, an explicit mapping \mathcal{F} does not always exist. In practice, the mapping is performed implicitly by choosing a suitable kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ for the data points \mathbf{x}_i and \mathbf{x}_j . However, there is a problem when choosing the function $K(\mathbf{x}_i, \mathbf{x}_j)$, since not every function is guaranteed to give a valid feature space. One way of searching for a valid kernel function is to draw on Mercer's theorem [17] which states that any continuous symmetric function $K(\mathbf{x}_i, \mathbf{x}_j)$ that satisfies the positive semidefinite condition $\int_{X \times X} K(\mathbf{x}_i, \mathbf{x}_j) f(\mathbf{x}_i) f(\mathbf{x}_j) d\mathbf{x}_i d\mathbf{x}_j \geq 0$ is ensured to be a kernel for some valid feature space. This provides a flexible way of choosing the kernel mapping functions. In this paper, we use the Gaussian and the polynomial kernels in our experiments on point-set matching process because of their transformational invariance.

To extract the principal components of the mapped data, first a covariance matrix needs to be constructed for the data in the feature space \mathcal{F} . Suppose that the data $\{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ in space \mathcal{F} are centred; then the covariance matrix of the mapped data in this space is

$$\bar{C} = \frac{1}{m-1} \sum_{i=1}^m \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_i)^T.$$

Since the explicit mapping \mathcal{F} is possibly unknown, computing the covariance matrix directly is not feasible. Schölkopf et al. showed in Ref. [15] that by solving the eigen-equation $m\lambda a = K\lambda$ in which the eigenvalues are $m\lambda$, the p_{th} feature vector, corresponding to the projection of the p_{th} feature point on the eigenspace, takes the form

$$\langle v^p, \Phi(\mathbf{x}) \rangle = \frac{1}{\sqrt{\lambda^p}} \sum_{i=1}^m \alpha_i^p k(\mathbf{x}_i, \mathbf{x}),$$

which can be further simplified to [18]

$$\langle v^p, \Phi(\mathbf{x}) \rangle = \frac{1}{\sqrt{\lambda^p}} (K \alpha^p)_n = \sqrt{\lambda^p} \alpha_n^p. \quad (2)$$

To generalize the method to non-centred data, the kernel function K becomes [15,18] $K' = (I - ee^T)K(I - ee^T)$ where $e = M^{-1/2}(1, 1, \dots, 1)^T$. In the case when more than one rigid component is present in the data point-set, the data need to be centred onto their respective subpart centre of movement. Thus, the mean value of each data group in the feature space \mathcal{F} needs to be computed and subtracted from $\Phi(\mathbf{x}_i)$ in the covariance matrix above. We make this separation using the label probabilities. For the group with label λ , the mean position (i.e. subgroup centre) is given by

$$\mu_\lambda = \frac{1}{\sum_i P(\theta_i = \lambda)} \sum_i \Phi(\mathbf{x}_i) P(\theta_i = \lambda) \quad \text{for each } \lambda \in \Omega.$$

Let $\tilde{\Phi}(\mathbf{x}_i) = (\Phi(\mathbf{x}_i) - \sum_\lambda \mu_\lambda P(\theta_i = \lambda))$; then the covariance matrix of centred data is given by

$$\tilde{C} = \frac{1}{m-1} \sum_{i=1}^m \tilde{\Phi}(\mathbf{x}_i) \cdot \tilde{\Phi}(\mathbf{x}_i)^T,$$

where

$$\begin{aligned} \tilde{\Phi}(\mathbf{x}_i) \cdot \tilde{\Phi}(\mathbf{x}_i)^T &= K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{\lambda \in \Omega} \frac{P(\theta_i = \lambda)}{\sum_j P(\theta_j = \lambda)} \\ &\times \sum_j P(\theta_j = \lambda) K(\mathbf{x}_i, \mathbf{x}_j) \\ &- \sum_{\lambda \in \Omega} \frac{P(\theta_i = \lambda)}{\sum_k P(\theta_k = \lambda)} \sum_k P(\theta_k = \lambda) K(\mathbf{x}_k, \mathbf{x}_i) \\ &+ \sum_{\lambda \in \Omega} \frac{P^2(\theta_i = \lambda)}{\sum_j P(\theta_j = \lambda) \sum_k P(\theta_k = \lambda)} \\ &\times \sum_j \sum_k P(\theta_j = \lambda) P(\theta_k = \lambda) K(\mathbf{x}_k, \mathbf{x}_j). \end{aligned} \quad (3)$$

Based on their transformational invariances, two kernel functions, the Gaussian kernel and the polynomial kernel, are of interest in this work. The Gaussian kernel has the form defined in Eq. (1). Since it is based on the Euclidean distance between two feature points, it is invariant to the similarity transformation. The polynomial kernel has the form $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + c)^d$, where c and d are constants ($d \neq 0$). This kernel captures the directionality of the data which is a useful property for correspondence matching. However, the scalar or dot product is not invariant under scaling, and so there is still a magnitude problem to be considered. To solve this problem, one approach is to normalise the scaled and truncated eigenvectors. Another method is to scale both of the two eigenvector matrices by the eigenvalue matrix of the model data-set. This is based on the following matrix property. Suppose that two eigen-decomposable matrices A and B are related by the scalar multiplier s , i.e. $A = s \cdot B$. Let the

eigen-decompositions of the two matrices be $A = U_A D_A U_A^T$ and $B = U_B D_B U_B^T$, where U_A and U_B form the orthonormal bases for the two matrices. When these conditions are satisfied then $D_A = s \cdot D_B$. In this work, we use this property to overcome the scale problem.

4. Label process

In the computer vision literature, one of the most extensively studied approaches to the consistent labelling problem involves the use of relaxation techniques. Relaxation labelling can either be an “offline” belief propagation process that distributes the previously learned labelling confidence over the entire feature set, e.g. Ref. [14], or an “online” learning process that learns the labelling information on the fly, e.g. Refs. [12,13]. In a discrete relaxation process (e.g. Ref. [19]), initially each node is assigned all possible labels. During the iterative relaxation procedure, inconsistent labels are discarded until a final consistent label distribution is obtained. In the continuous or probabilistic case, each node is assigned an initial weight or probability distribution. Iteratively, the label probabilities or weights are updated, again until a consistent distribution is reached. However, whichever labelling process is used, the performance depends critically on the compatibility coefficients used and the support function used to combine evidence in the iterative process. In Ref. [13], a dictionary is used, and in Ref. [14] the compatibility coefficients are represented as a vector which is learned offline. These definitions of the compatibility coefficients are not suitable for our point-correspondence problem. However, our compatibility model does share some properties in common with the compatibility vector in Ref. [14].

The labelling process that is being developed here is an evidence combining one that propagates label constraints. Our label consistency model is derived from the one of the feature point-sets, which we refer to as the model. We hence learn the label compatibility information from the model point-set, before attempting to match it against the data point-set. Our first step is to collect label information from the model point-set, and apply the learned label compatibility model in the second step of the process which involves assigning consistent point labels to the “data” point-set.

We are interested in matching two point-sets $\mathbf{y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ and $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$. The individual point-sets are characterised using a Gaussian proximity matrix. For the points with positions \mathbf{x}_i and \mathbf{x}_j the element of the matrix W was defined earlier in Eq. (1).

4.1. Label compatibility information

Our aim is to develop a relational description of the point-sets using information concerning point proximity and a label compatibility matrix. The compatibility matrix $R \in \mathbb{R}^{L \times L}$ is of dimension $L \times L$ and embodies knowledge of the

number of rigid components, i.e. labels, in each image, and the semantic constraints between each pair of object-labels. The matrix has elements

$$R_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ come from the same rigid part,} \\ -1 & \text{otherwise.} \end{cases}$$

This definition restricts the nodes to give total positive support to the nodes in the same group and to contribute a negative support to nodes outside the group. The proximity constraint is also acquired from the model image. We assume that in any two consecutive image frames, the relative position of the rigid components of the object under study will not change significantly.

4.2. Label update formula

The label probabilities for the data point-set are updated iteratively commencing from a set of initial values. With the label compatibility information learned from the model point-set, we update the label probability for each point according to the support from its neighbourhood. Let us denote the neighbourhood for the point \mathbf{x}_i and its k closest points by $N_i = \{\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_k}\}$. Here we use the Euclidean distance between the points \mathbf{x}_i and \mathbf{x}_j to define the neighbourhood. With these ingredients the support from the neighbourhood for the label assignment λ_i to point \mathbf{x}_i is

$$S_{i,\lambda_i} = \frac{\exp\{\sum_{j \in N_i} \sum_{\lambda_j \in \Omega} P(\theta_j = \lambda_j) R(\lambda_i, \lambda_j) W_{ij}\}}{\sum_{\lambda_i \in \Omega} \exp\{\sum_{k \in N_i} \sum_{\lambda_k \in \Omega} P(\theta_k = \lambda_k) R(\lambda_i, \lambda_k) W_{ik}\}}, \quad (4)$$

where $R(\lambda_i, \lambda_j)$ are the elements of the label compatibility matrix R which measure the compatibility of the label pair λ_i and λ_j . Here the elements of the proximity matrix W are defined using Eq. (1) and are used to weight the label-support.

Having defined the support equation, the label probabilities are iteratively updated using the formula

$$P^{(n+1)}(\theta_i = \lambda) = \frac{P^{(n)}(\theta_i = \lambda) + \mu S_{i,\lambda}^{(n)}}{\sum_{\lambda_i \in \Omega} (P^{(n)}(\theta_i = \lambda) + \mu S_{i,\lambda}^{(n)})}, \quad (5)$$

where μ is a constant parameter and n is the iteration index.

5. Matching

In this section, we describe our point matching algorithm and detail how it uses kernel PCA and spectral graph theory. We develop two different algorithms. The first of these is designed to work with single objects undergoing rigid motion, while the second is designed to work with objects which have rigid components that undergo articulated motion relative to one another. We commence by developing an efficient point pattern matching process for feature points under rigid motion. Here we require a kernel function that

will capture the transformational invariances of the object movement, and allow the feature points to be embedded into a lower dimensional feature space in a manner that provides a basis for one-to-one correspondence matching. Our second contribution is to develop a point pattern matching method that can be applied to feature point-sets from objects that undergo articulated movement. We do this by incorporating label consistency constraints into the rigid matching process.

5.1. Rigid case

The idea underpinning the use of kernel spectral methods is to first represent the transformation invariant relationships between the feature points in terms of a proximity matrix for each feature point-set. Then correspondences are located by using the eigen-decompositions of the matrix pairs. That is, we first choose an appropriate kernel function to extract the transformation invariant feature properties from the point-sets. By performing kernel PCA on the data-set we extract a transformationally invariant basis set for the feature points. For the feature points from rigid motion, the Gaussian and polynomial kernel functions satisfy our requirements. The procedure for performing rigid matching by kernel PCA is described in the pseudo-code given in Table 1.

This matching algorithm is motivated by the approach described by Shapiro and Brady [3]. In Ref. [3], a proximity matrix W is first constructed for each image with the matrix elements given by $W_{ij} = \exp\{-d_{ij}^2/2\sigma^2\}$, where d_{ij}^2 is the Euclidean distance between points \mathbf{x}_i and \mathbf{x}_j , and σ is an adjustable parameter. Shapiro and Brady explain this as the mapping of the original two-dimensional data to a higher-dimensional space, and thus capture the structural arrangement of the feature points. They then perform eigendecomposition on matrix W to obtain its eigenvalues and eigenvectors. For each point-set, a new modal matrix is constructed with the eigenvectors sorted in descending eigenvalue order as its columns. The rows of the matrix are then considered

as the projections of the feature points into the eigenspace. When the data-sets are of different size, only the first M leading eigenvectors from each data-set are used where M is the size of the smaller data-set. To make the algorithm more robust, Shapiro and Brady also suggest that the eigenvalues should be used to scale the corresponding eigenvectors so as to place more emphasis on the more significant eigenvectors. When the eigenvalues are involved, this approach is similar to kernel PCA. When compared with the kernel PCA method described above, it is clear that the Shapiro and Brady method is a special case where the data in the mapped space have a mean zero and use the Gaussian as the kernel function.

5.2. Articulated case

In the case of matching feature point-sets resulting from articulated motion, the above matching method cannot be used directly since the motion of each individual rigid component changes the relative positions of the feature points from the different components. Thus, the overall structural change is large. Our idea is to use label consistency constraints to construct a modified proximity matrix. In particular, we propose to use the weighted adjacency matrix \tilde{W} for each point-set in which each element of the matrix is computed using the formula

$$\tilde{W}_{ij} = \sum_{l=1}^L P(\theta_i = l)P(\theta_j = l)W_{ij}, \quad (6)$$

where W_{ij} is defined in Eq. (1). The resulting matrix is then subjected to the kernelisation procedure outlined in Eq. (3) and its eigen-decomposition computed. The mapping of the feature vectors $\tilde{\mathbf{y}}_j$ and $\tilde{\mathbf{x}}_i$ is thus computed by using Eq. (2) for the respective modal and data point-sets; that is,

$$m\lambda\alpha = W\lambda, \quad \tilde{\mathbf{x}}_i = \sqrt{\lambda^i}\alpha_n^i,$$

and a similar procedure is applied to $\tilde{\mathbf{y}}_j$. The next step is to compute the association matrix $M_{ij} = \exp(-d_{ij}^2/\sigma)$, where $d_{ij}^2 = \|\tilde{\mathbf{y}}_i - \tilde{\mathbf{x}}_j\|^2$ is the distance of the point pairs. Let us denote the label agreement of the point pair \mathbf{y}_i and \mathbf{x}_j by $P(\theta_j = \lambda, \theta_i = \lambda, \forall \lambda \in \Omega)$. The association of the two feature vectors is further gated by this constraint:

$$\tilde{M}_{ij} = P(\theta_j = \lambda, \theta_i = \lambda, \forall \lambda \in \Omega)M_{ij}. \quad (7)$$

The correspondences are defined as the most similar node pairs. That is, for each node \mathbf{x}_i in the data point-set, the correspondence in the model set is the node \mathbf{y}_j that has the largest association \tilde{M} . If we assume that the labels on each feature point are independent of one other, the consistency of the label assigned to point \mathbf{x}_i and the label assigned to \mathbf{x}_j is given by

$$P(\theta_i = \lambda, \theta_j = \lambda, \forall \lambda \in \Omega) = \sum_{\lambda=1}^L P(\theta_i = \lambda)P(\theta_j = \lambda). \quad (8)$$

Table 1
Algorithm 1

Kernel spectral matching : Algorithm 1	
(1)	Compute the proximity matrices: $W_{ij}^1 = K(\mathbf{y}_i, \mathbf{y}_j)$, $W_{ij}^2 = K(\mathbf{x}_i, \mathbf{x}_j)$,
(2)	$W^1 = (I - ee^T)W^1(I - ee^T)$ $W^2 = (I - ee^T)W^2(I - ee^T)$
(3)	Eigen-decomposition: $W^1 = U^1 A^1 (U^1)^T$; $W^2 = U^2 A^2 (U^2)^T$;
(4)	Compute the projections: $\mathbf{y}' = U^1 \sqrt{(nA^1)}$ $\mathbf{x}' = U^2 \sqrt{(mA^2)}$
(5)	Correspondence for $\mathbf{x}'_i = \min_{\mathbf{y}_j, j=1, \dots, n} \text{dist}(\mathbf{x}'_i, \mathbf{y}'_j)$.

Table 2
Algorithm II

Kernel spectral matching : Algorithm II	
(1)	Initialise P, t, S_{old} ;
(2)	Learn the compatibility information from \mathbf{y} ;
(3)	Compute the Gaussian association matrix for \mathbf{x} ;
(4)	Run the labelling process, compute P^{new} ;
(5)	Use P^{new} to compute \bar{C}_{new} using Eq. (3) and \tilde{M} using (7);
(6)	Find for each $\mathbf{x}_i \in \mathbf{x}$ its correspondence $\mathbf{y}_j = \max_j \tilde{M}_{ij}$;
(7)	Compute S , $\text{diff} = S - S_{old}$; if $\text{diff} < \text{threshold}$ or $\text{iteration} < t$ return; else update P using the matching results. end
(8)	Go to step 3.

The matching process is an interactive one in which at each step new label probabilities are incorporated to improve matching. Since an increasing number of correspondences are found, the value of the quantity $S = \sum_i \exp(\|\mathbf{x}_i - \mathbf{y}_i\|_F / 2\sigma^2)$, where \mathbf{x}_i and \mathbf{y}_i are the correspondence pair from data point-set and model point-set, respectively, will increase, and ultimately reach a maximum value. Thus we use this quality as one of our stopping criteria of the matching process. The other is a predefined iteration number. The matching process is summarised in the pseudocode listed in Table 2.

6. Experimental results

In this section we present our experimental evaluation of both the rigid and articulated matching methods. Our experiments are performed with both synthetic and real world data. We also compare the proposed algorithms with Shapiro and Brady's algorithm [3], Scott and Longuet-Higgins' Algorithm [2], and embedded point-set matching using the multi-dimensional scaling. MDS is also a method widely used for data dimension reduction, and is also based on eigenvalues and eigenvectors of a dissimilarity matrix [20]. It attempts to preserve the pairwise relationships between the data points while mapping the data into a low-dimensional space. The experiments are performed using the classical MDS in which the Euclidean distance is taken as the dissimilarity measure.

The experiments focus on the performance of the algorithms when the data are subjected to transformations and contain uncertainties such as outliers, random position jitter and also small deformations.

6.1. The data

The experiments are performed on both synthesised data and real world data sets. The experimental designs are as follows:

- (1) *Synthetic data*: Here we assume that the point-sets are subject to a two-dimensional affine transformation. Given a point-set $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ sampled from a

rigid object, a synthetic data-set $X' = sRX + \mathbf{t}$ is generated for testing the algorithms, where s is a scaling parameter, \mathbf{t} is the translation vector, and

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

is the 2D rotation matrix. In our experiments with Algorithm I (the rigid variant), the rotation angle $\theta = \frac{10}{180} \times \pi$. In the experiments for Algorithm II (the articulated variant), the transformation parameters are set as $s = 0.8$, $\mathbf{t} = (10, 15)T'$ and $\theta = (20/180)\pi$, and in the third component, $s = 1.2$, $\mathbf{t} = (10, 15)T'$ and $\theta = (30/180)\pi$.

- (2) *Real data*: Here we use two sequences for Algorithm I. The first of these is a sequence of infrared images of a hand shown in Fig. 1 which is used as an example where the geometric deformations are small. The second sequence is the CMU house sequence [6] shown in Fig. 2, which is used to study the effects of matching point-sets of different size and where there is significant positional jitter. The image pairs in Fig. 3 are used to experiment with Algorithm II on scenes where there is articulated object movement. The top image shows two rectangular objects that move relative to each other on the ground-plane. The bottom image shows a pair of spectacles, where one of the limbs moves.
- (3) *Noisy data*: Here Gaussian noise is added to the data-set to test the robustness of the algorithm. The data are synthesised in the following way. First synthetic point position jitter is synthesised by generating a matrix $D \sim N(\mu, \Sigma)$ whose elements are Gaussian random variables with mean μ and covariance matrix Σ . The point jitter is added to the matrix of feature point-set positions for the second point set X_2 using the equation $X_2 = X_2 + D$.
- (4) *Data-sets with different size*: To simulate structural errors we delete a controlled fraction of the feature points from the data point-set. This is done in two different ways. Firstly l consecutive points, where l starts from 1, to the integer number most close to 12% of the data-set size, are deleted to simulate occlusion. Secondly, l points are deleted from random locations to simulate the effect of segmentation errors. For the CMU house sequence and the glasses sequence, the feature points are extracted using a corner detector which produces errors and hence the point-sets are of different sizes. For instance, in frames 01, 02, 03, 04, 05, and 06 of the CMU sequence displayed in Fig. 2, the sizes of the point-sets are 30, 32, 32, 30, 30, and 32, respectively.

6.2. Results

To compare the performance of the kernel approaches when deformations are present, experiments are performed on synthetically generated data where a 2D translation,

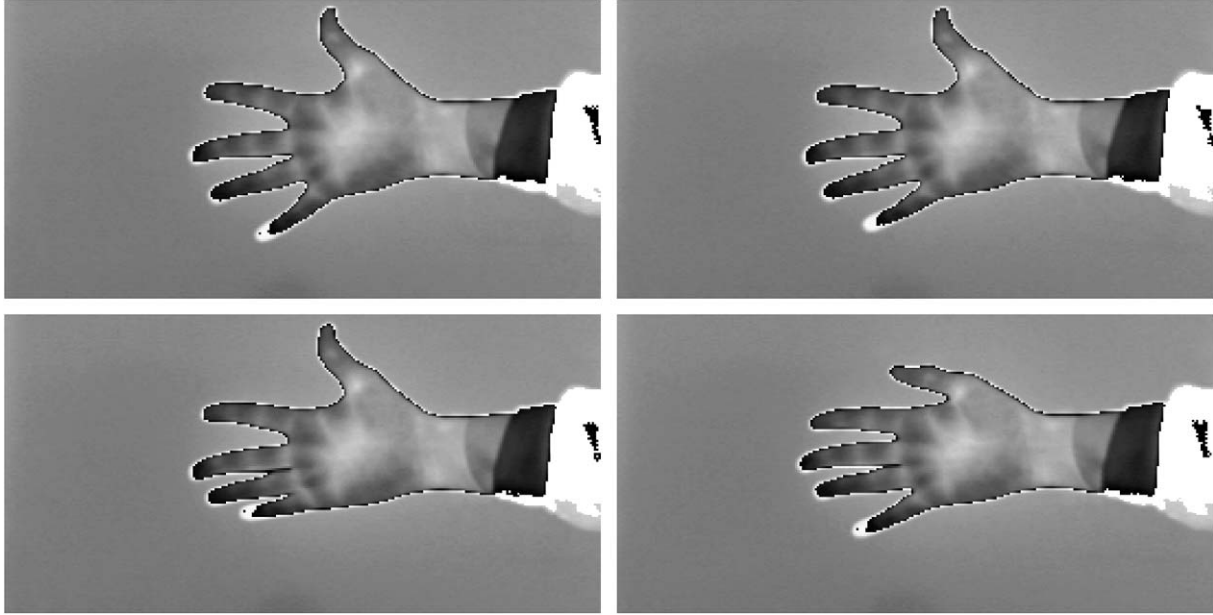


Fig. 1. Hand image data (from left to right, top to bottom: frame 08, 09, 11, 25).

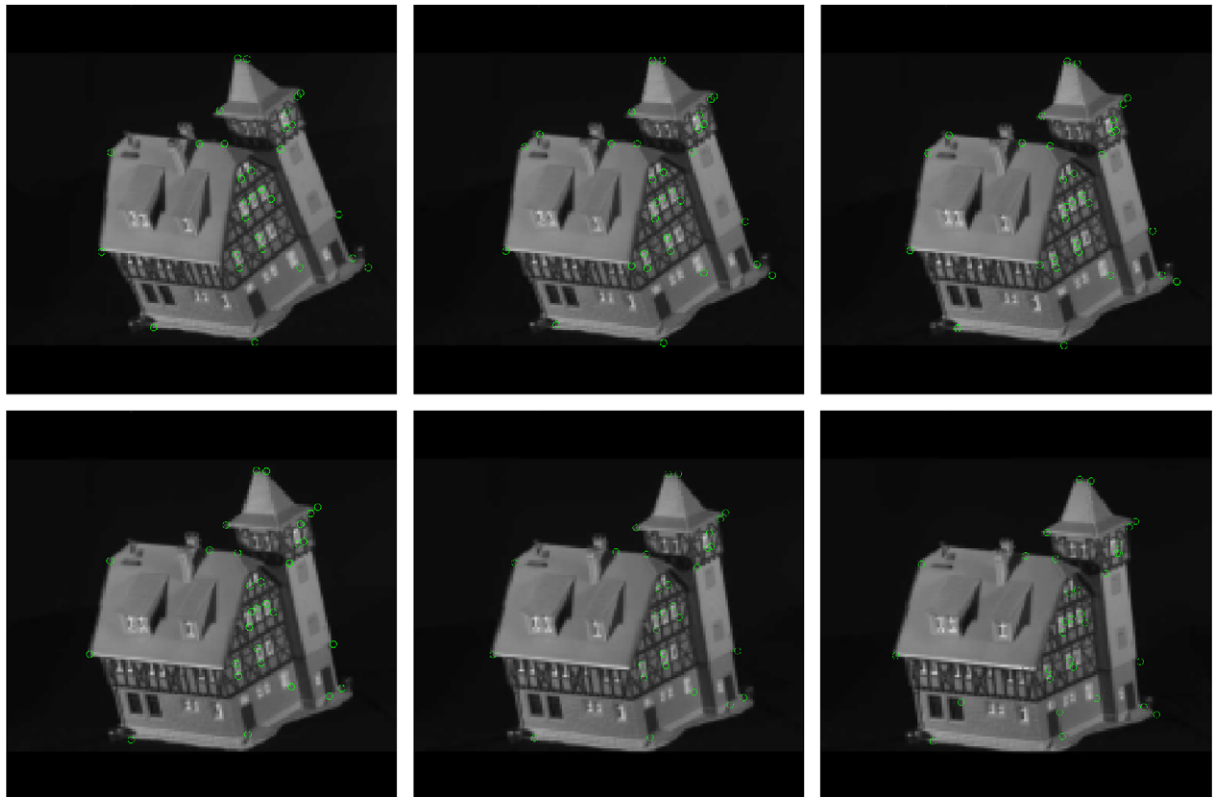


Fig. 2. CMU house data (from left to right, top to bottom: frame 01, 02, 03, 04, 10).

rotation and isoscaling are added. The effect of missing points and random point position jitter in terms of the 2D Gaussian random matrices with different covariance matrices as described above are also tested.

6.2.1. The σ value

When using the Gaussian kernel, the choice of the σ value significantly affects the performance of the algorithm when the data points contain significant uncertainties. We com-

mence by investigating the effect of varying the parameter. Intuitively, we expect the value of σ to be strongly dependent on the pairwise distances between the feature

points. Here we use the formula

$$\sigma = h\bar{d}_{ij}^2, \quad \bar{d}_{ij}^2 = \sum_{ij} d_{ij}^2 / (n^2 - n)$$

to estimate the parameter, where h is a scalar parameter, d_{ij} is the Euclidean distance between the points i and j , and n is the size of the feature point-set. For four different feature point-set pairs, in Fig. 4 we show the effect of varying the parameter σ on the percentage of correctly matched points. The curve marked with crosses is for the kernel PCA method with a Gaussian kernel and the curve marked with circles is for the Shapiro and Brady method. In all four cases, the best performance of the kernel PCA method is better than that delivered by Shapiro and Brady.

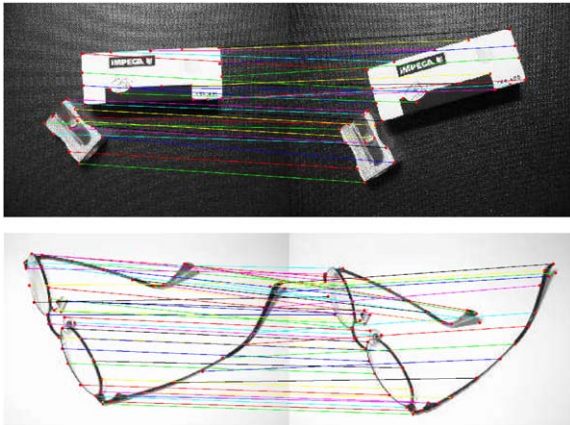


Fig. 3. Articulated matching result: data-set 3 and 5.

6.2.2. Label process

We now turn our attention to experiments which focus on the performance of the label process for both synthetic and

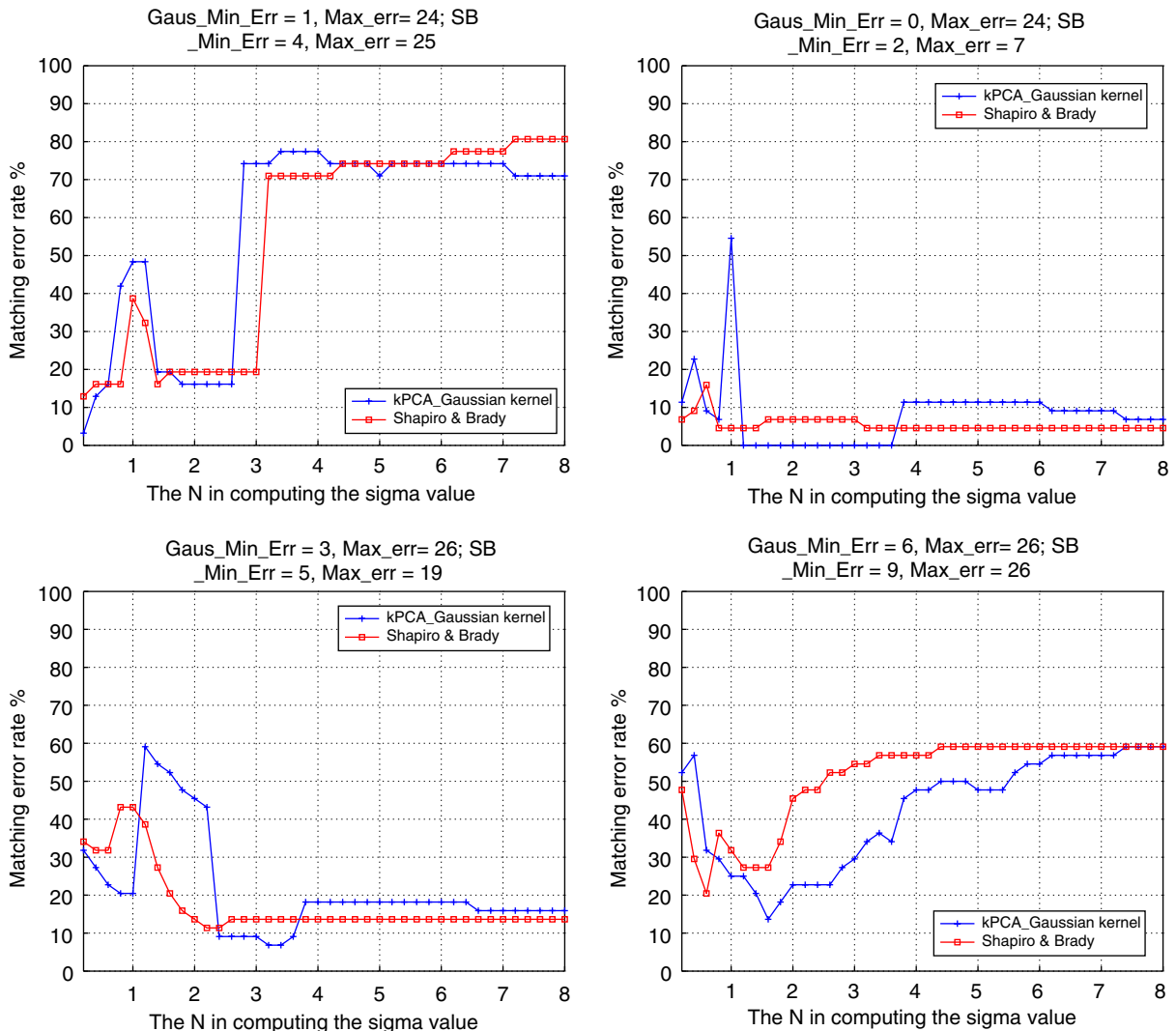


Fig. 4. Effects of different σ value (upper-left: data-set 3; upper-right: hand 08/09; lower-left: hand 08/11; lower-right: hand 08/25).

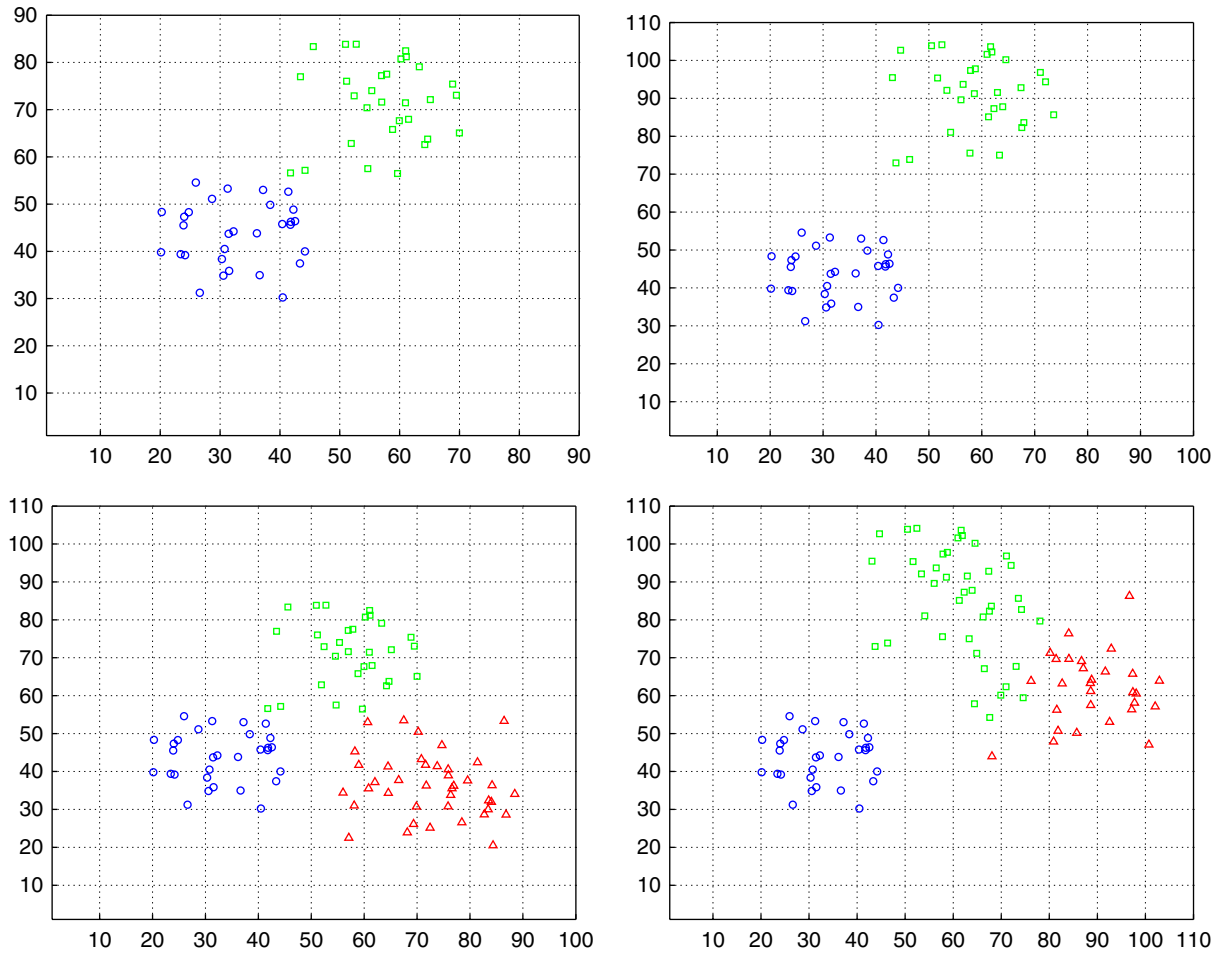


Fig. 5. Synthetic data with 2 and 3 rigid components in motion, and their labelling result. Top: data-set pair 2; bottom: data-set pair 1.

real image point-sets. The initial values of the label probabilities have an effect on the number of iterations and the rate of convergence of the method. Fig. 5 shows the result of labelling random point-sets. The initial label probabilities are assigned uniformly. The different colours correspond to the labels assigned to the points. As the clusters of points become more and more overlapped, then the number of labelling errors increase.

6.2.3. Matching

We commence by considering the case of rigid correspondence. We first experiment on the single component synthetic data generated using the methods described above. For this data-set, our algorithm for rigid motion, Shapiro and Brady's method, and Scott and Longuet-Higgins's method give correspondences that are 100% correct, while the MDS approach results in a 5% error rate. The effect of missing points and random position jitter for rigid point matching are shown in Figs. 6 and 7. In the case of random jitter, the experimental results are the averages of 100 runs for each covariance matrix. For missing points, the experimental results are obtained by consecutively

deleting a predetermined number of points from the data point-set.

The results of applying the algorithms to real data-sets are shown in Figs. 6, 7, and Table 3. Here we show the fraction of correct correspondence errors as a function of the number of missing points (performed in the same way as for the synthetic data). In Figs. 6 and 7 the different curves are for the different algorithms. As the number of deleted points increases, then the best performance is obtained when kernel PCA and a polynomial kernel are used. The poorest results are obtained with kPCA and a Gaussian kernel in the experiments on synthetic data-sets. But in the experiments on real image data, the kPCA with a Gaussian kernel performs better than the approaches in Refs. [20,2,3]. In all of the experiments using Shapiro and Brady's method, the eigenvalues are used to normalise the corresponding eigenvectors in order to improve the matching results. This gives results that are comparable to MDS.

In Fig. 8 we compare the results of using the Gaussian and polynomial kernels on point-sets with increasing jitter. There is relatively little to distinguish their performance. Figs. 9, 10 and 11, respectively, show the fraction of correct matches as

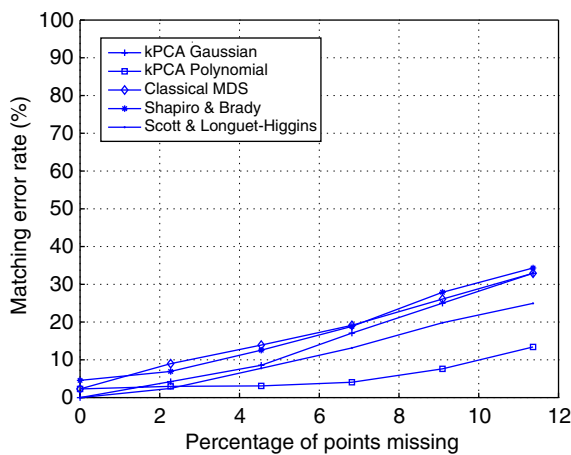
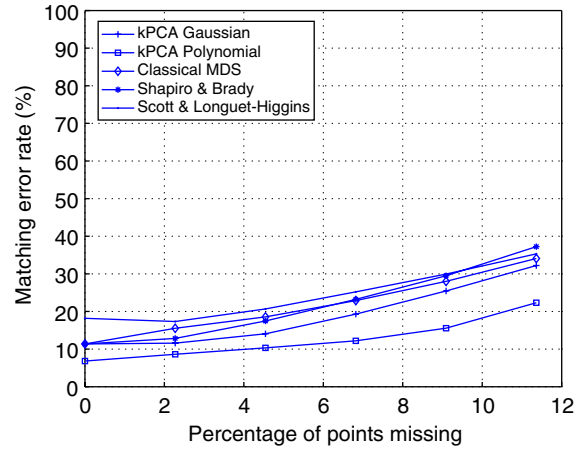
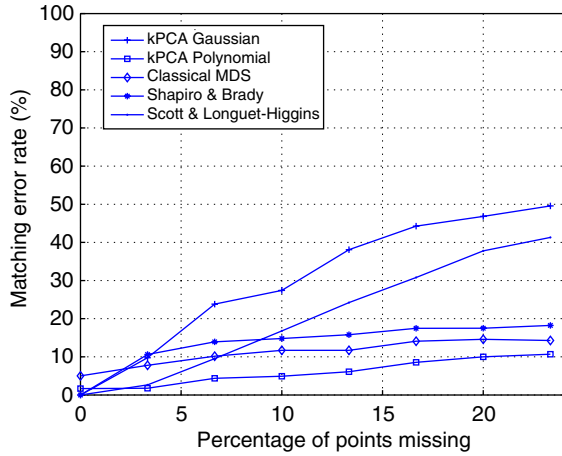


Fig. 6. Matching results (top: synthetic data; bottom: hand 08 and 09).

Fig. 7. Matching results (top: Algorithm I, hand 08 and 11, consecutive deletions; bottom: Algorithm II, randomly deleted points).

function of the fraction of random point deletions, the standard deviation of the point-jitter and the fraction of points occluded. Occlusion is simulated by deleting a fraction of consecutive feature points from the data point-set (as described previously). Random position jitter is simulated by adding randomly generated position errors sampled from a 2D Gaussian distribution to the data point-set (as in the rigid matching experiments). The different curves in the plots are for different numbers of components (labels). In

each case as the noise increases, then so the error also increases. The numbers used in these plots are summarised in Table 4.

From these experiments, it is clear that the kernel PCA approach gives encouraging results when compared with the approaches of Shapiro and Brady [3], Scott and Longuet-Higgins [2], and the MDS method. Moreover, the kernel method is less sensitive to noise than the alternatives.

The main computational overheads of the algorithms described are associated with constructing the kernel matrices, and computing their eigenvalues and eigenvectors. This may prove burdensome for very large data-sets. The complexity of computing the eigensystem of the kernel matrix is cubic in the number of points. If the feature point-sets are of a moderate size (e.g. less than 10^3 points), then computing the full eigensystem for a matrix of size 930×930 takes less than half a minute on a desktop PC with an AMD Athlon 2000 CPU and 256 MB RAM. The computation of the kernel matrix can be performed with time complexity that is polynomial in the size of the point-sets. The number of iterations required is also an important consideration. Algorithm II usually requires only three iterations. For the rigid point correspondence matching problem, our algorithm is non-iterative.

Table 3
Matching results (Algorithm I, Numbers of errors)

Frames	Hand data				CMU house				
	08/25	09/11	09/25	11/25	01/02	01/03	01/04	01/05	01/06
KPCA, Gaussian	6	4	4	11	2	4	2	2	7
KPCA, polynomial	5	7	6	12	4	5	3	5	13
MDS	35	5	26	27	5	5	25	25	28
Shapiro and Brady	9	6	8	17	3	5	2	2	9
SLH	4	3	5	10	7	6	3	7	9

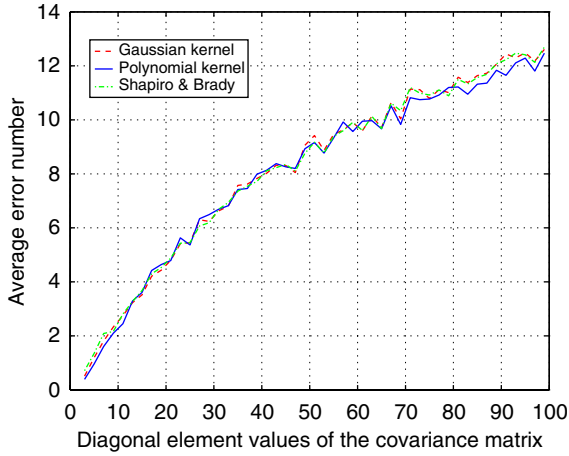


Fig. 8. Effects of Gaussian random point jitter.

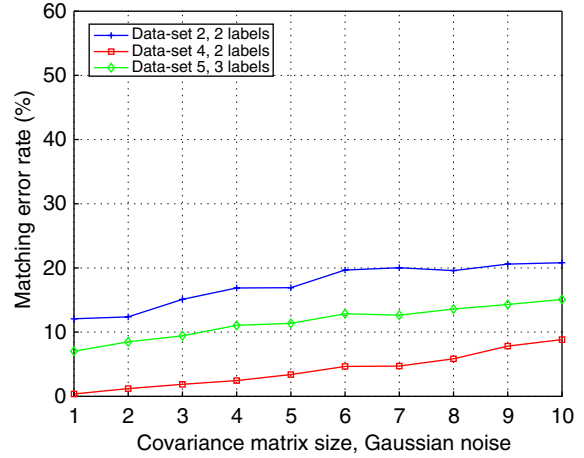


Fig. 10. Effects of random point jitter.

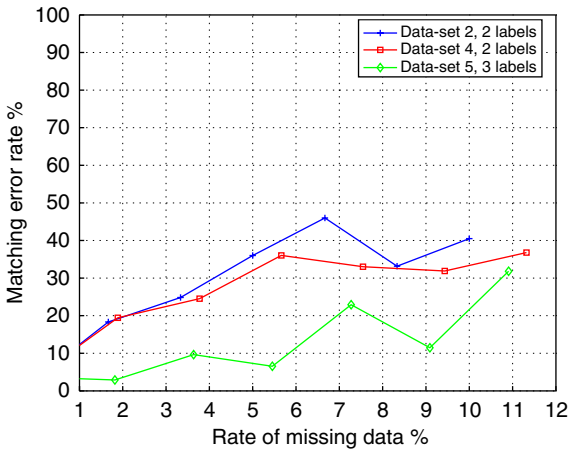


Fig. 9. Effects of random point deletions.

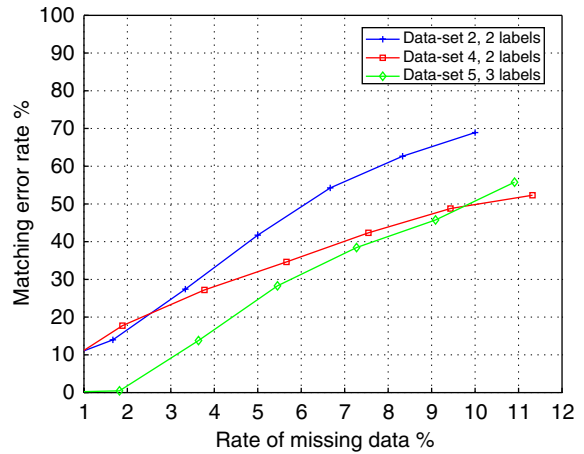


Fig. 11. Effect of occlusion.

Table 4
Matching and labelling results (Algorithm II, Gaussian kernel, error%)

Data-set	Number of points	Number of labels	Number of label information	Articulated matching (1) ^a	Articulated matching (2) ^b	Labelling
1	100	3	95	93	16	7
2	60	2	53.33	13.33	13.33	0
3	31	2	35.48	0	0	0
4	55	2	18.18	7.27	3.64	1.82
5	53	3	45.28	81.13	3.77	7.55
6	10	3	10	0	0	0

^aResults obtained based on the label information from the label process.

^bResults obtained when correct label information is assumed.

7. Conclusions

In this paper we have made two contributions. First, we have explored the use of kernel PCA with a polynomial

kernel function for finding correspondences between two feature point-sets. The relationship of the methods with Shapiro and Brady’s correspondence method [3] is established and discussed. Experimental results reveal that the

method offers performance advantages over a number of alternative methods. Here the polynomial kernel proves to be the most stable for point-sets of different sizes, and even in worst cases it gives a tolerable error rate. The performance of our algorithm is also comparable to the approaches described in Refs. [5,6,7]. One weakness of the Gaussian kernel is the selecting of the width parameter σ . In Ref. [3], the value is chosen manually. In this paper, we use a heuristic formula based on the inter-point pairwise Euclidean distance matrix to compute σ automatically. The kernel functions used in this paper are possibly not the best functions for extracting invariant properties from the feature point-sets. Other kernels may be used in the matching process to improve the results.

Our second contribution has been to extend the kernel method to articulated point-sets. Here we show how label compatibility coefficients can be used to refine the computation of the kernelised proximity matrix. The method improves the correspondence process when there are different moving components of a scene.

Our future plans are as follows. First, we aim to explore the use of different kernels, and in particular ones that allow the invariances of the transformation between point-sets to be captured in a natural way. Second, we plan to do more work on the label process. One possibility which has a natural assonance with the kernel method is to use the heat equation and its spectral solution to model the evolution of label probabilities with time. Work aimed at investigating these points is in hand and will be reported in due course.

References

- [1] F.R.K. Chung, Spectral graph theory, American Mathematical Society, Regional Conference Series in Mathematics, vol. 92, 1997.
- [2] G.L. Scott, H.C. Longuet-Higgins, An algorithm for associating the features of two images, Proc. R. Soc. London Ser. B 244 (1991) 21–26.
- [3] L.S. Shapiro, J.M. Brady, Feature-based correspondence—an eigenvector approach, Image and Vision Computing 10 (1992) 283–288.
- [4] M. Pilu, A direct method for stereo correspondence based on singular value decomposition, IEEE CVPR 1997.
- [5] M. Carcassoni, E.R. Hancock, Spectral correspondence for point pattern matching, Pattern Recognition 36 (2003) 193–204.
- [6] M. Carcassoni, E.R. Hancock, Correspondence matching with modal clusters, IEEE Trans. PAMI 25 (12) (2003).
- [7] B. Luo, E.R. Hancock, Matching point-sets using Procrustes alignment and the EM algorithm, BMVC 1999.
- [8] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, Training models of shape from sets of examples, in: Proceedings BMVC, 1992.
- [9] C. Tomasi, T. Kanade, Shape and motion from image streams under orthography—a factorization method, Technical Report TR-92-1270, Cornell University, 1992.
- [10] S. Pappu, S. Gold, A. Rangarajan, A framework for non-rigid matching and correspondence, Adv. Neural Inform. Process. Systems 8 (1996) 795–801.
- [11] R.C. Wilson, E.R. Hancock, Structural matching by discrete relaxation, IEEE Trans. PAMI 19 (6) (1997).
- [12] A. Rosenfeld, R. Hummel, S. Zucker, Scene labeling by relaxation operations, IEEE Trans. Systems, Man Cybernet. 6 (1976) 420–433.
- [13] J. Kittler, E.R. Hancock, Combining evidence in probabilistic relaxation, Int. J. Pattern Recognition Artif. Intell. 3 (1) (1989) 29–51.
- [14] M. Pelillo, M. Refice, Learning compatibility coefficients for relaxation labeling processes, IEEE Trans. PAMI 16 (9) (1994) 933–945.
- [15] B. Schölkopf, A.J. Smola, K.R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (1998) 1299–1319.
- [16] S. Mika, B. Schölkopf, A. Smola, K.R. Müller, M. Scholz, G. Rätsch, Kernel PCA and de-noising in feature spaces, in: NIPS 1998.
- [17] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
- [18] J. Ham, D.D. Lee, S. Mika, B. Schölkopf, A kernel view of the dimensionality reduction of manifolds, Max Planck Institute for Biological Cybernetics, Technical Report TR-110, 2003.
- [19] D.L. Waltz, Generating semantic descriptions from drawings of scenes with shadows, MIT AI Lab Technical Report 271, 1972.
- [20] T.F. Cox, M.A.A. Cox, Multidimensional Scaling, Chapman & Hall, London, 1994.
- [21] T. Caelli, S. Kosinov, An eigenspace projection clustering method for inexact graph matching, IEEE Trans. Pattern Anal. Machine Intell. 26 (4) (2004).
- [22] S. Sclaroff, A.P. Pentland, Modal matching for correspondence and recognition, IEEE Trans. Pattern Anal. Machine Intell. 17 (6) (1995).
- [23] S. Umeyama, An eigendecomposition approach to weighted graph matching problem, IEEE Trans. Pattern Anal. Machine Intell. 10 (5) (1988).
- [24] W. Christmas, J. Kittler, M. Petrou, Structural matching in computer vision using probabilistic relaxation, IEEE Trans. PAMI 17 (1995) 749–764.
- [25] H. Chui, A. Rangarajan, A new point matching algorithm for non-rigid registration, Comput. Vision Image Understanding 89 (2003) 114–141.
- [26] M. Pelillo, K. Siddiqi, S.W. Zucker, Matching hierarchical structures using association graphs, IEEE Trans. Pattern Anal. Machine Intell. 21 (11) (1999) 1105–1120.
- [27] M.E. Tipping, C.M. Bishop, Mixtures of probabilistic principal component analyzers, Neural Comput. 11 (2) (1999) 443–482.

Further reading

Hong Fang Wang got her B. Eng. and M.Sc. degrees from Xi'an Jiaotong University, China, and University of Edinburgh, UK, respectively. Before she started her M.Sc. study in Edinburgh, she worked about 6 years as a computer engineer in China. She is currently a Ph.D. student in the Department of Computer Science, University of York.

Edwin Hancock studied physics as an undergraduate at the University of Durham and graduated with honours in 1977. He remained at Durham to complete a Ph.D. in the area of high-energy physics in 1981. Following this he worked for 10 years as a researcher in the fields of high-energy nuclear physics and pattern recognition at the Rutherford–Appleton Laboratory (now the Central Research Laboratory of the Research Councils). During this period he also held adjunct teaching posts at the University of Surrey and the Open University. In 1991 he

moved to the University of York as a lecturer in the Department of Computer Science. He was promoted to Senior Lecturer in 1997 and to Reader in 1998. In 1998 he was appointed to a Chair in Computer Vision.

Professor Hancock now leads a group of some 15 faculty, research staff and Ph.D. students working in the areas of computer vision and pattern recognition. His main research interests are in the use of optimisation and probabilistic methods for high- and intermediate level vision. He is also interested in the methodology of structural and statistical pattern recognition. He is currently working on graph-matching, shape-from-X, image data-bases and statistical learning theory. His work has found applications in areas such as radar terrain analysis, seismic section analysis, remote sensing and medical imaging. Professor Hancock has published some 90 journal papers and 350 refereed conference publications. He was awarded the Pattern Recognition Society medal in 1991 for the best paper to be published in the journal *Pattern Recognition*. The journal also awarded him an outstanding paper award in 1997.

Professor Hancock has been a member of the Editorial Boards of the journals *IEEE Transactions on Pattern Analysis and Machine Intelligence*, and *Pattern Recognition*. He has also been a guest editor for special editions of the journals *Image and Vision Computing* and *Pattern Recognition*, and he is currently a guest editor of a special edition of *IEEE Transactions on Pattern Analysis and Machine Intelligence* devoted to energy minimisation methods in computer vision. He has been on the programme committees for numerous national and international meetings. In 1997 he established a new series of international meetings on energy minimisation methods in computer vision and pattern recognition. He was awarded a Fellowship of the International Association for Pattern Recognition in 2000.