Dimension-Free Affine Shape Matching Through Subspace Invariance

Zhaozhong Wang and Han Xiao
Image Processing Center, Beihang University
Beijing 100191, China
zwang@buaa.edu.cn, xiaohan@sa.buaa.edu.cn

Abstract

This paper proposes an affine invariant matching algorithm for shape correspondence problems in arbitrary dimensions. Formulating shapes by configuration matrices of landmarks, and using the fact that subspaces (e.g. range spaces) of these matrices are invariant to affine transformations, the shape correspondence is modelled as a permutation relation between orthogonal projection matrices of the subspaces. Then the matching result is solved by an efficient factorization procedure for rank-deficient matrices. The algorithm is compact, fast, and independent of dimensions. Experimental results for 1D, 2D and 3D matchings of synthetic and real data are provided, which demonstrate potential applications of the algorithm to shape analysis, and to other related problems like wide baseline stereo matching and range data registration.

1. Introduction

Matching algorithms in computer vision require invariance properties, e.g. invariant to similarity or affine transformations. Shape is an useful concept to formulate invariance, and is generally defined as geometric information invariant under Euclidean similarity transformations [9, 13]. The definition can also be generalized to include affine invariance. Shape matching is a critical step in shape analysis, see [15] for a recent review and evaluation.

Shape matching methods based on landmarks are typically the multi-step approaches, e.g., using centralization, normalization and principle-axis alignment to remove translation, scaling and rotation, respectively. There may exist additional steps to matching deformations using techniques like the thin-plate spline. For example, the work of Dalal *et al.* [7] applies such a multi-step approach for initially 3D matching of shapes against similarity transformations. Ha and Moura [11] recover an intrinsic shape which is invariant to affine-permutation distortions, and use the steps of centering, reshaping, reorientation and sorting. Ho *et al.* [12] also propose a multi-step affine matching algorithm; their method is only applicable to 2D point matching because their theory is based on complex number. There are other kinds of shape matching algorithms, such as the MDL (minimum description length) [8] and the shape context algorithm [4], but these algorithms are more complex.

Designing a single-step or compact algorithm for affine shape matching is attractive. If additionally, the algorithm is applicable for shape matching under arbitrary dimensions, it will be more useful. This paper tries to achieve these objects. The shapes to be matched will be represented by configuration matrices of landmarks. The affine invariance in matching is established via invariance of range spaces of the configuration matrices; such an invariance has been studied in [3] and other recent work. Orthogonal projection matrices of the range spaces are proved in this paper to be permutation similar. Then the unknown permutation, which encodes the landmark correspondence, is solved based on the *QR factorizations with column pivoting* for the rank-deficient orthogonal projection matrices. Such kind of factorizations remove the full set of affine transformations at the same time. The algorithm is fast, independent to dimensions, and robust to perturbations such as perspective transformations. As typical shape matching methods, shape landmarks need to be known in advance (manually marked or automatically extracted) for our algorithm, and should be of the same number for different shapes. The landmark coordinates are the only input to the algorithm.

Some existing correspondence algorithms seem to be dimension free, and are based on matrix decompositions as well. For instance, Scott and Longuet-Higgins [17] design an inter-image proximity matrix between landmarks using Gaussian functions, then solve for correspondences via the singular value decomposition (SVD). Shapiro and Brady [18] improve the SVD method by constructing intra-image proximity matrices for the individual point sets, then using the eigenvalue decomposition to recover the matching. Their algorithm is further generalized to spectral graph methods [6] and graph matching methods [5]. Theoretically, the QR factorization with column pivoting we used is faster than the eigenvalue decomposition and SVD, and

more importantly, it is applicable to rank-deficient matrices to result in an unique solution for correspondence.

2. Correspondence from Subspace Invariance

We consider in general the shape matching problem in $k-1$ dimensional real space $\mathbb{R}^{k-1}$, where $k = 2, 3, \cdots$. A point $\mathbf{x}_i$ in this space can be expressed by a k-D homogenous coordinate as $\mathbf{x}_i = [\, x_{1i} \; x_{2i} \; \cdots \; x_{k-1,i} \; 1 \,]^T$. Assuming that the landmark set of shape consists of n points, we can construct the following *configuration matrix* [9]:

$$X = [\, \mathbf{x}_1 \; \mathbf{x}_2 \; \cdots \; \mathbf{x}_n \,]^T \in \mathbb{R}^{n \times k}. \tag{1}$$

We assume that $n \geq k$, and the matrix X is of full rank, i.e. $\text{rank}(X) = k$.

The landmark set $\{ \mathbf{x}_1 \; \mathbf{x}_2 \; \cdots \; \mathbf{x}_n \}$ can be distorted by any affine transformation through the matrix multiplication XA, where

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,k-1} & 0 \\ \cdots & \cdots & \cdots & \cdots \\ a_{k-1,1} & \cdots & a_{k-1,k-1} & 0 \\ a_{k,1} & \cdots & a_{k,k-1} & 1 \end{bmatrix} \in \mathbb{R}^{k \times k} \tag{2}$$

is an affine matrix. Our matching algorithm should be invariant to the transformation A. To this end, consider the range space $\mathcal{R}(X)$ [10],

$$\mathcal{R}(X) = \{ \mathbf{u} \in \mathbb{R}^n : \mathbf{u} = X\mathbf{v}, \, \mathbf{v} \in \mathbb{R}^k \},$$

and $\mathcal{R}(XA)$. It has been shown in the literature (e.g. in [3]) that, for any nonsingular affine matrix A, we have

$$\mathcal{R}(XA) = \mathcal{R}(X), \tag{3}$$

i.e., the range space of X is invariant under affine transformations. Eq. (3) is the starting-point of our theory.

Given configuration matrices of two shapes, the matching of landmarks between the shapes can be modelled as the reordering of rows of one matrix so that its rows have the same order as corresponding rows in another matrix. Such a process can be performed by left-multiplying the matrix using an permutation matrix. In this way, we assume that a configuration matrix Y is transformed from X by any affine transformation and permutation as

$$Y = PXA,$$

where $P \in \mathbb{R}^{n \times n}$ is a permutation matrix. We need to solve for the unknown P from the measured matrices X and Y, under any distortion of A.

We now give an approach different from existing methods to solve the above problem. Firstly, using Eq. (3) we know that $\mathcal{R}(PXA) = \mathcal{R}(PX)$. We then define a subspace for any matrices $X \in \mathbb{R}^{n \times k}$ and $P \in \mathbb{R}^{n \times n}$ as

$$P[\mathcal{R}(X)] := \{ \mathbf{x} : \mathbf{x} = P\mathbf{u}, \, \mathbf{u} \in \mathcal{R}(X) \}. \tag{4}$$

For any $\mathbf{x} \in \mathcal{R}(PX)$, there is $\mathbf{v} \in \mathbb{R}^k$ such that $\mathbf{x} = PX\mathbf{v}$. Clearly $X\mathbf{v} \in \mathcal{R}(X)$, so $\mathbf{x} \in P[\mathcal{R}(X)]$ according to Eq. (4). Conversely, for any $\mathbf{x} \in P[\mathcal{R}(X)]$, there is an $\mathbf{u} \in \mathcal{R}(X)$ such that $\mathbf{x} = P\mathbf{u}$, and in turn $\mathbf{u} = X\mathbf{v}$ for some $\mathbf{v} \in \mathbb{R}^k$. Therefore $\mathbf{x} = PX\mathbf{v}$ and $\mathbf{x} \in \mathcal{R}(PX)$. Thus we proved that $\mathcal{R}(PX) = P[\mathcal{R}(X)]$. So there is

$$\mathcal{R}(PXA) = P[\mathcal{R}(X)] \tag{5}$$

for any nonsingular matrices $A \in \mathbb{R}^{k \times k}$ and $P \in \mathbb{R}^{n \times n}$.

Eq. (5) reveals a subspace transformation under permutations, bypassing the effect of the affine matrix A. But this equation is inconvenient to use in practice. Fortunately, it is known that a subspace uniquely corresponds to an orthogonal projection matrix [10]. Let $\Pi_X, \Pi_Y \in \mathbb{R}^{n \times n}$ be the orthogonal projection matrices of the subspaces $\mathcal{R}(X)$ and $\mathcal{R}(Y) = P[\mathcal{R}(X)]$, respectively. Then we know that $\Pi_X = VV^T$, where $V \in \mathbb{R}^{n \times k}$ is a matrix whose column vectors form an orthonormal basis for $\mathcal{R}(X)$. As the matrix P is orthogonal, the columns of the matrix PV form an orthonormal basis for the subspace $P[\mathcal{R}(X)]$. Consequently,

$$\Pi_Y = P\Pi_X P^T. \tag{6}$$

This indicates that, under any affine transformation and permutation, the two orthogonal projection matrices Π_X and Π_Y are *permutation similar*. Eq. (6) is affine invariant, which may be used to solve for the unknown permutation.

The orthogonal projection matrices Π_X and Π_Y are typically rank deficient as $n \geq k$, so general matrix decomposition methods like the eigenvalue decomposition and SVD fail to solve the permutation from Eq. (6). The equation may be solved by integer quadratic programming or concave minimization [14], but these methods are NP-hard. We shall propose an efficient solution below.

3. Solution and Algorithm

Eq. (6) will be solved based on a rank-deficient matrix decomposition method, called the *QR factorization with column pivoting* [10]. This method has the complexity $O(kn^2)$, while the eigenvalue decomposition and SVD have the complexity $O(n^3)$. It is known that a typical QR factorization for a matrix M is given by $M = QR$ with Q an orthogonal matrix and R a upper-triangular matrix. But if M is rank deficient, the diagonal entries of R contain zeros and the factorization is non-unique. In this case, the QR factorization with column pivoting is useful. For the rank-deficient matrices Π_X and Π_Y, we have

$$\Pi_X \Gamma = [Q \; Q'] \begin{bmatrix} R & R' \\ 0 & 0 \end{bmatrix}, \tag{7a}$$

$$\Pi_Y \tilde{\Gamma} = [\tilde{Q} \; \tilde{Q}'] \begin{bmatrix} \tilde{R} & \tilde{R}' \\ 0 & 0 \end{bmatrix}, \tag{7b}$$

where $\Gamma, \tilde{\Gamma} \in \mathbb{R}^{n \times n}$ are permutation matrices, $Q, \tilde{Q} \in \mathbb{R}^{n \times k}$ and $Q', \tilde{Q}' \in \mathbb{R}^{n \times (n-k)}$ are matrices with orthonormal columns, $R, \tilde{R} \in \mathbb{R}^{k \times k}$ are upper-triangular matrices and $R', \tilde{R}' \in \mathbb{R}^{k \times (n-k)}$ nonzero matrices. We shall derive briefly the relation between the above two factorizations. For convenience we assume that diagonal elements of R and $\tilde{R}$ are positive.

The factorizations in Eq. (7) are generally performed using the *Householder transformation* [10]:

$$H = I - 2\mathbf{v}\mathbf{v}^T / (\mathbf{v}^T \mathbf{v}),$$

where $\mathbf{v} \in \mathbb{R}^n$, and $I \in \mathbb{R}^{n \times n}$ is the identity matrix. Let $\mathbf{e}_i$, $i = 1, \cdots, n$, be the i-th column vector of I. For any nonzero vector $\mathbf{x} \in \mathbb{R}^n$, set $\mathbf{v} = \mathbf{x} \pm \|\mathbf{x}\|_2 \mathbf{e}_1$, then $H\mathbf{x} = \mp \|\mathbf{x}\|_2 \mathbf{e}_1$. That is, the Householder matrix H zeroes all components of the vector $\mathbf{x}$, except the first one. We can multiply H by $\mathbf{e}_1^T$ to obtain the first row of H as

$$\mathbf{e}_1^T H = \mp \frac{1}{\|\mathbf{x}\|_2} \mathbf{x}^T. \tag{8}$$

Assume that for some k we have computed

$$\Theta_{k-1}^T := H_{k-1} \cdots H_1, \quad \Gamma_{k-1} := T_1 \cdots T_{k-1},$$

such that

$$\Theta_{k-1}^T \Pi_X \Gamma_{k-1} = R^{(k-1)} = \begin{bmatrix} R_{11}^{(k-1)} & R_{12}^{(k-1)} \\ 0 & R_{22}^{(k-1)} \end{bmatrix}, \tag{9}$$

where $R_{11}^{(k-1)}$ is a nonsingular and upper-triangular matrix. Each matrix T_i corresponds to a mapping t_i on the set $\{1, \cdots, n\}$ to interchanging columns of Π_X, so the matrix Γ_{k-1} corresponds to the mapping $\gamma_{k-1} = t_{k-1} \circ \cdots \circ t_1$.

Now suppose that the column partitioning of $R_{22}^{(k-1)}$ is given by

$$R_{22}^{(k-1)} = [\mathbf{z}_k^{(k-1)}, \cdots, \mathbf{z}_n^{(k-1)}],$$

and let $m \geq k$ be the index such that

$$\|\mathbf{z}_m^{(k-1)}\|_2 = \max\{\|\mathbf{z}_k^{(k-1)}\|_2, \cdots, \|\mathbf{z}_n^{(k-1)}\|_2\}. \tag{10}$$

If we assume that none of the two norms of $\mathbf{z}_j^{(k-1)}$ are equal, then the index m is unique. Note that if $\mathrm{rank}(\Pi_X) = k-1$, then this maximum is zero and we are finished. Otherwise, let T_k be a $n \times n$ permutation interchanging columns m and k of $R^{(k-1)}$ and determine a matrix H_k such that

$$R^{(k)} = H_k R^{(k-1)} T_k,$$

where

$$H_k = \mathrm{diag}(I_{k-1}, H_k'), \tag{11}$$

with I_{k-1} the $(k-1) \times (k-1)$ identity matrix and H_k' the $(n-k+1) \times (n-k+1)$ Householder matrix. In other words, the permutation matrix T_k moves the largest column in $R_{22}^{(k-1)}$ to the lead position and H_k' zeroes all of its sub-diagonal components.

From Eq. (8) we know that the first row of $H_1 = \Theta_1^T$ is given by

$$\mathbf{e}_1^T \Theta_1^T = \mathbf{e}_1^T H_1 = \frac{1}{r_{11}} \mathbf{u}_{\gamma_1^{-1}(1)}^T, \tag{12}$$

where $r_{11} = \|\mathbf{u}_{\gamma_1^{-1}(1)}\|_2$ is the first diagonal element of $R_{11}^{(k-1)}$ as we have assumed its positivity, and $\mathbf{u}_{\gamma_1^{-1}(1)}$ is a column vector of Π_X interchanged from the column $\gamma_1^{-1}(1) = t_1^{-1}(1)$ to the column 1.

For the matrix Π_Y we have a formula similar to Eq. (9) with matrices $\tilde{\Theta}_{k-1}^T := \tilde{H}_{k-1} \cdots \tilde{H}_1$ and $\tilde{\Gamma}_{k-1} := \tilde{T}_1 \cdots \tilde{T}_{k-1}$. We have supposed that $\Pi_X = [\mathbf{u}_1, \cdots, \mathbf{u}_n]$, now let $\Pi_Y = [\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_n]$. From Eq. (6) we get

$$\tilde{\mathbf{u}}_j = P \mathbf{u}_{p(j)}, \quad j = 1, \cdots, n, \tag{13}$$

where $p(\cdot)$ is an index mapping on $\{1, \cdots, n\}$ corresponding to the permutation P. That is, we first move a column $\mathbf{u}_i$ of Π_X to the position $p^{-1}(i)$ using P^T, then permute the elements of $\mathbf{u}_{p^{-1}(i)}$ using P to obtain $\tilde{\mathbf{u}}_{p^{-1}(i)} = \tilde{\mathbf{u}}_j$.

It is easy to obtain from Eq. (13) that $\|\tilde{\mathbf{u}}_j\|_2 = \|\mathbf{u}_{p(j)}\|_2$, so if $\tilde{\mathbf{u}}_j$ in Π_Y is moved to the first position, $\mathbf{u}_{p(j)}$ in Π_X must also be moved to the first position, provided that none of any two column vectors in Π_X (or Π_Y) have the same norm. That is

$$\tilde{\mathbf{u}}_{\tilde{\gamma}_1^{-1}(1)} = P \mathbf{u}_{p(\tilde{\gamma}_1^{-1}(1))} = P \mathbf{u}_{\gamma_1^{-1}(1)},$$

where $\tilde{\gamma}_1 = \tilde{t}_1$ is the index mapping corresponding to $\tilde{\Gamma}_1 = \tilde{T}_1$ for Π_Y. This equation indicates that

$$\tilde{r}_{11} = \|\tilde{\mathbf{u}}_{\tilde{\gamma}_1^{-1}(1)}\|_2 = \|\mathbf{u}_{\gamma_1^{-1}(1)}\|_2 = r_{11}.$$

In addition, we know that the first column of $\tilde{\Theta}_1$ is

$$\tilde{\Theta}_1 \mathbf{e}_1 = \frac{1}{\tilde{r}_{11}} \tilde{\mathbf{u}}_{\tilde{\gamma}_1^{-1}(1)} = \frac{1}{r_{11}} P \mathbf{u}_{\gamma_1^{-1}(1)} = P\Theta_1 \mathbf{e}_1.$$

Generalizing the above process recursively, we can obtain

$$\tilde{\Theta}_i \mathbf{e}_i = P\Theta_i \mathbf{e}_i, \quad i = 1, \cdots, k. \tag{14}$$

Additionally, using Eq. (11) we can deduce that

$$\Theta_k \mathbf{e}_i = \Theta_i \mathbf{e}_i, \quad \tilde{\Theta}_k \mathbf{e}_i = \tilde{\Theta}_i \mathbf{e}_i, \quad i = 1, \cdots, k-1. \tag{15}$$

If $\mathrm{rank}(\Pi_X) = \mathrm{rank}(\Pi_Y) = k$, the Householder iterations stop after k steps and

$$\tilde{\Theta}_k = [\tilde{Q} \ \tilde{Q}'], \quad \Theta_k = [Q \ Q'].$$

Then

$$\begin{aligned} \tilde{Q} &= [\tilde{\Theta}_k \mathbf{e}_1 \ \tilde{\Theta}_k \mathbf{e}_2 \ \cdots \ \tilde{\Theta}_k \mathbf{e}_k] \\ &= [\tilde{\Theta}_1 \mathbf{e}_1 \ \tilde{\Theta}_2 \mathbf{e}_2 \ \cdots \ \tilde{\Theta}_k \mathbf{e}_k] \\ &= [P\Theta_1 \mathbf{e}_1 \ P\Theta_2 \mathbf{e}_2 \ \cdots \ P\Theta_k \mathbf{e}_k], \end{aligned}$$

X			Y			Q			$\tilde{Q}$			P				
977.6	668.1	1	47.1	86.8	1	-0.0145	-0.9172	0.0000	0.0762	-0.5131	-0.1917	0	1	0	0	0
960.6	484.1	1	154.6	141.6	1	0.0642	-0.3741	-0.3156	0.1415	0.1639	-0.6669	0	0	0	1	0
1258.6	278.1	1	311.1	277.0	1	-0.1596	0.0630	-0.7494	0.9752	0.0000	0.0000	0	0	0	0	1
907.6	323.0	1	373.6	272.0	1	0.1707	0.1217	-0.5821	-0.0162	-0.8424	0.0000	1	0	0	0	0
25.0	529.1	1	334.6	118.5	1	0.9701	0.0000	0.0000	-0.1513	-0.0152	-0.7201	0	0	1	0	0

Table 1. Example of computational data for 2D shape matching from real noisy landmarks [in Fig. 2(a)] with $n = 5$. X and Y are 5×3 configuration matrices of landmarks. Q and $\tilde{Q}$ are the matrices in Eq. (16). P is the computed permutation matrix using Eq. (18). P reveals the correspondence between the row vectors of Q and $\tilde{Q}$, i.e., the correspondence between the landmarks in X and Y.
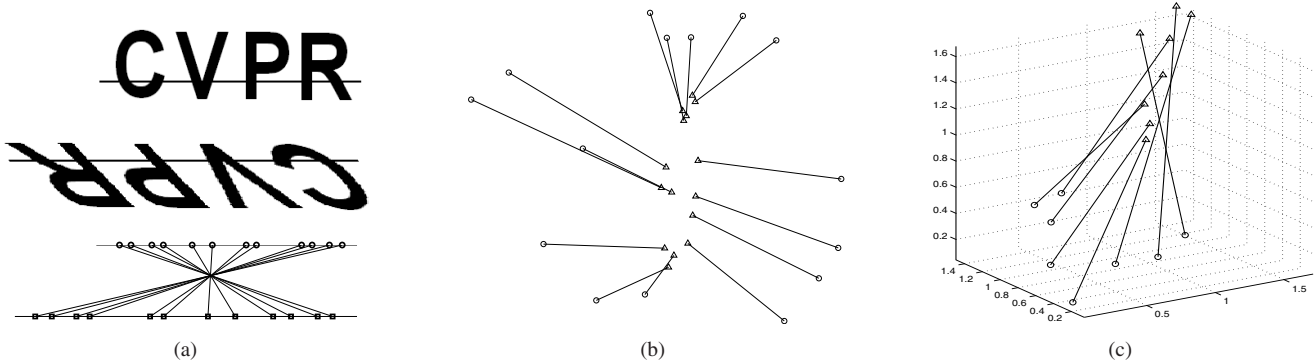

Figure 1. Examples of 1D, 2D and 3D matching for synthetic data. (a) Matching in 1D: The top two are synthetic images with marked corresponding lines. The below is the matching between the two lines, where points on each line are the intersections between the line and the characters' edges. (b) and (c) are matchings in 2D and 3D, between point sets marked as circles and triangles, respectively.

thus we finally have

$$\tilde{Q} = PQ. \tag{16}$$

Eq. (16) can be used to solve for the unknown permutation P by comparing the row vectors of $\tilde{Q}$ and Q. This equation gives a rigorous relation in the noiseless case.

In the above derivations, the condition that the column norms of $R_{22}^{(k-1)}$ are unequal is crucial; if some of the column norms are equal, then the shape has some kind of symmetry, e.g. the rotational or reflectional symmetry. In these cases, the matching is not unique. The condition that the diagonals of R and $\tilde{R}$ are positive can be reduced to the condition that corresponding diagonal entries of R and $\tilde{R}$ have the same sign. Typical QR routines with column pivoting (as those in [2]) cannot guarantee this condition, so a sign correction step may be necessary.

Let the matrices R and $\tilde{R}$ in Eq. (7) have diagonal entries r_{ii} and $\tilde{r}_{ii}$, $i = 1, \cdots, k$, respectively. Define a diagonal matrix $S \in \mathbb{R}^{k \times k}$ whose diagonal entries s_{ii} is given by

$$s_{ii} = \text{sign}(r_{ii}\tilde{r}_{ii}), \quad i = 1, \cdots, k, \tag{17}$$

where $\text{sign}(\cdot)$ is the signum function. Thus the matrix product $S\tilde{R}$ makes the diagonal elements of $\tilde{R}$ having the same sign as those of R. To ensure the entire factorization unchanged, we should simultaneously change the matrix $\tilde{Q}$ to $\tilde{Q}S$ since $\tilde{Q}SS\tilde{R} = \tilde{Q}\tilde{R}$. This sign correction process is much easier than those used in the spectral methods [18].

Now we can summarize our algorithm, which is designed based on Eq. (16) but is applicable to noisy landmarks. The input is the configuration matrices X and Y, and the output is the permutation matrix P:

1. Compute the orthogonal projection matrices Π_X and Π_Y from X and Y, respectively.

2. Compute the QR factorization of Π_X and Π_Y, to obtain Q and $\tilde{Q}$ respectively, as in Eq. (7).

3. Construct a diagonal matrix S from Eq. (17), then set $\tilde{Q} = \tilde{Q}S$.

4. Let $b_{ij} := \|\mathbf{e}_i^T \tilde{Q} - \mathbf{e}_j^T Q\|_2$, then the permutation matrix $P = [p_{ij}]$, $i, j = 1, \cdots, n$, is given by

$$p_{ij} = \begin{cases} 1, & b_{ij} \leq b_{il}, \forall l, \text{ and } b_{ij} \leq b_{kj}, \forall k, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

This algorithm is simple and robust. Next we illustrate its performance with experiments.

4. Experimental Results

Experimental results for synthetic and real data under 1D, 2D and 3D are provided. The shape matching under 2D and 3D are widely used in computer vision, while the 1D matching is seldom. A typical example of 1D matching is the stereo matching between epipolar lines, where the landmark set on a line can be viewed as an "1D shape". Affine transformations in 1D reduce to translations and scalings

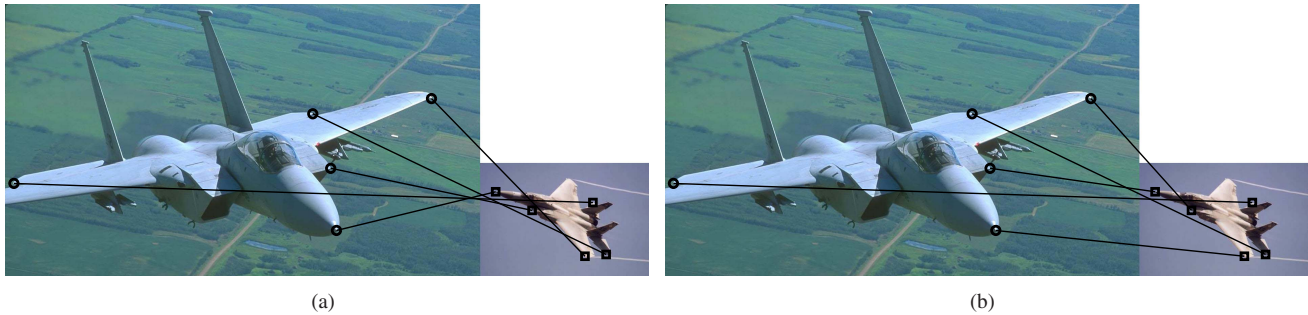(a)   (b)

Figure 2. Examples of 2D matching for real images, formed by perspective projections, with large degree of rotation and scaling. The big image is with size 1400×800, and the small image size is 498×334. (a) The matching result using our algorithm based on the QR factorization with column pivoting. (b) The matching result of the eigenvalue decomposition algorithm in [18].

(a)   (b)

Figure 3. Examples of 2D matching for a image pair with large degree of rotation in (a), and for a stereo pair with wide baseline in (b).

on the line; reflection in 1D can be taken into account as well. Fig. 1(a) gives an example for 1D matching under these transformations using our algorithm. Figs. 1(b) and (c) provide results for 2D and 3D matching, where landmarks and affine transformations are generated randomly.

For the matching of real 2D images and 3D data, landmarks are selected manually in random order. Then the correspondences are estimated using the proposed algorithm. Fig. 2(a) gives a matching result under perspective projections, where the images are searched on the Web. Our algorithm approximates well the perspective transformations, even if there are large degree of rotation and scaling. Table 1 provides computational data for this example. We also compare our algorithm with the eigenvalue decomposition approach [18] in Fig. 2(b). This method is not suitable for matching under anisotropic scaling or large rotation, and in addition, it needs empirical parameters; while our algorithm is parameter free.

Fig. 3 contains more experimental results for real image data of the archive [1]. The capability of the algorithm for wide baseline matching is demonstrated here. Fig. 4(a) shows a 3D matching of surface meshes, which may be applied to the field of 3D morphing. Note that the matching in Fig. 4(a) has a reflection effect, which is caused by the reflectional symmetry of landmarks. The matching result would be unique if the symmetry vanished, similar to the landmark distribution in Fig. 2(a). Fig. 4(b) depicts a matching for 3D range data. This would be used in range data registration to remove affine transformations in an initial step of deformable matching.

5. Conclusion and Future Work

In our affine invariant shape matching algorithm, we derived from the invariance of range spaces of configuration matrices, to obtained the correspondence model as a relation of permutation similarity between orthogonal projection matrices. Then we solved the matching problem based on the QR factorizations with column pivoting for the rank-deficient orthogonal projection matrices. The factorizations are simpler and faster than the eigenvalue decomposition and SVD. The algorithm is dimension free and parameter free, i.e., it can be used for matching in arbitrary integer dimensions, and there are no parameters to be set. The algorithm may be used for shape analysis to remove affine transformations, for wide baseline stereo matching or for range data registration.

A major drawback of the algorithm is that landmarks need to given in advance and should be of same number for different shapes; the algorithm does not include steps to extract and select landmarks. So future work should aim to select proper landmarks from automatically extracted features. We also need to perform perturbation analysis of the algorithm in theory, with respect to random noise and more general deformable deformations. The matrix factorization method in this paper may provide a new representa-

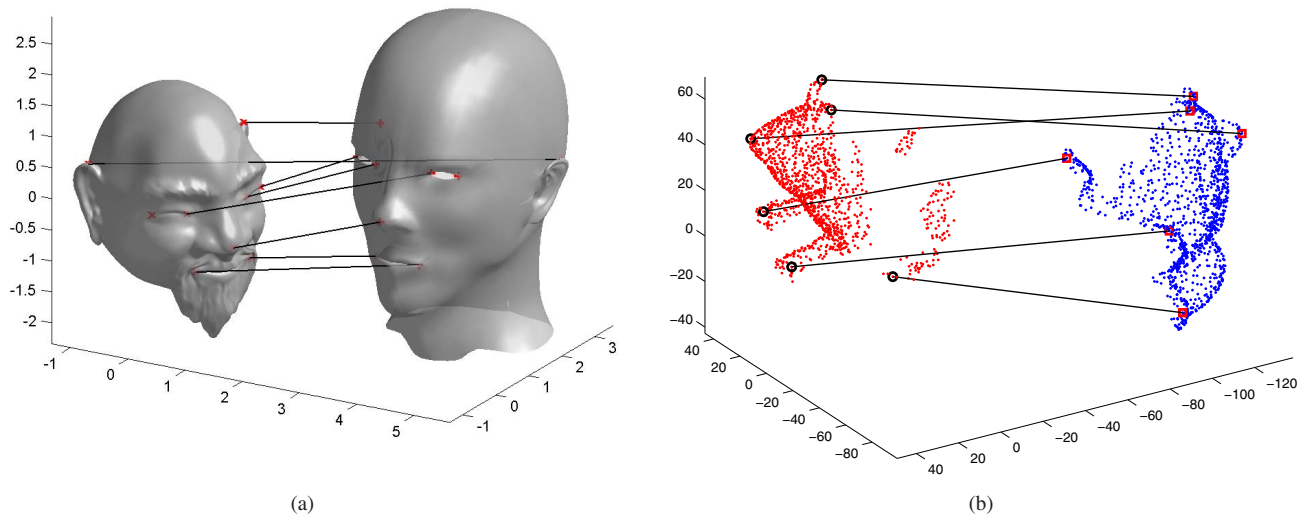(a)   (b)

Figure 4. Examples of 3D matching. (a) The matching between synthetic 3D models. (b) The matching for real range data of frog, which is obtained from [16].

tion scheme for shapes (e.g. using the matrix Q to replace the matrix X), which needs to be studied further.

Acknowledgement

References

[1] Image archive of computational vision group at caltech. http://www.vision.caltech.edu.

[2] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK User's Guide*. SIAM Publications, Philadelphia, 3rd edition, 1999.

[3] E. Begelfor and M. Werman. Affine invariance revisited. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages II: 2087–2094, 2006.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[5] T. Caetano, T. Caelli, D. Schuurmans, and D. Barone. Graphical models and point pattern matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(10):1646–1663, 2006.

[6] M. Carcassoni and E. Hancock. Spectral correspondence for point pattern matching. *Pattern Recognition*, 36:193–204, 2003.

[7] P. Dalal, B. Munsell, S. Wang, J. Tang, K. Oliver, H. Ninomiya, X. Zhou, and H. Fujita. A fast 3d correspondence method for statistical shape modeling. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[8] R. Davies, C. Twining, T. Cootes, J. Waterton, and C. Taylor. A minimum description length approach to statistical shape modeling. *IEEE Trans. Medical Imaging*, 21(5):525–537, 2002.

[9] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. Wiley, New York, 1998.

[10] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[11] V. Ha and J. Moura. Affine-permutation invariance of 2-d shapes. *IEEE Trans. Image Processing*, 14(11):1687–1700, 2005.

[12] J. Ho, M. Yang, A. Rangarajan, and B. Vemuri. A new affine registration algorithm for matching 2d point sets. In *IEEE Workshop on Applications of Computer Vision*, pages 25–25, 2007.

[13] D. G. Kendall, D. Barden, T. K. Carne, and H. Le. *Shape and Shape Theory*. Wiley, New York, 1999.

[14] J. Maciel and J. Costeira. A global solution to sparse correspondence problems. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(2):187–199, 2003.

[15] B. Munsell, P. Dalal, and S. Wang. Evaluating shape correspondence for statistical shape analysis: A benchmark study. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(11):2023–2039, 2008.

[16] J. Salvi, C. Matabosch, D. Fofi, and J. Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25(5):578–596, 2007.

[17] G. Scott and H. Longuet-Higgins. An algorithm for associating the features of two images. *Proc. of Royal Society of London B*, 244:21–26, 1991.

[18] L. Shapiro and J. Brady. Feature-based correspondence: An eigenvector approach. *Image and Vision Computing*, 10(5):283–288, 1992.