A New Affine Registration Algorithm for Matching 2D Point Sets

Jeffrey Ho
CISE Department
University of Florida
Gainesvill, FL
jho@cise.ufl.edu

Ming-Hsuan Yang
Honda Research Institute
800 California St
Mountain View, CA
mhyang@ieee.org

Anand Rangarajan
CISE Department
University of Florida
Gainesvill, FL
anand@cise.ufl.edu

Baba Vemuri
CISE Department
University of Florida
Gainesvill, FL
vemuri@cise.ufl.edu

Abstract

We propose a novel affine registration algorithm for matching 2D point sets. Unlike many previously published work on affine point matching, the proposed algorithm does not require any optimization and in the absence of data noise, the algorithm will recover the exact affine transformation and the unknown correspondence. The two-step algorithm first reduces the general affine case to the orthogonal case, and the unknown rotation is computed as the roots of a low-degree polynomial with complex coefficients. The algebraic and geometric ideas behind the proposed method are both clear and transparent, and its implementation is straightforward. We validate the algorithm on a variety of synthetic 2D point sets as well as feature points on images of real-world objects.

1. Introduction

Matching points in 2D has been a classical problem in computer vision. The problem can be formulated in a variety of ways depending on the allowable deformations. For instance, the orthogonal and affine cases have been studied already awhile ago, e.g., [6][9]. Recent research activities have focused on non-rigid deformations, particularly those that can be locally modelled by a family of well-known analytic functions such as splines (e.g., [3]). In this paper, we study the classical problem of matching two point sets in $\mathbb{R}^2$ related by rigid (orthogonal) or affine transformations.

Let $\mathcal{P} = \{p_1, \cdots, p_k\}$ and $\mathcal{Q} = \{q_1, \cdots, q_k\}$ denote two collections of points in $\mathbb{R}^2$, and $\mathrm{N}_{\mathcal{Q}}(x)$ denote the minimal distance between the point x and points in $\mathcal{Q}$. The affine registration problem (which includes the rigid registration as a special case) asks for an affine transformation $\mathbf{A} = (A, t)$ that minimizes the following least-square error function:

$$\mathcal{E}(A, t) = \sum_{i=1}^{k} \mathrm{N}_{\mathcal{Q}}(Ap_i + t)^2. \tag{1}$$

In the above, $A \in \mathbf{GL}(2)$ is a non-singular 2×2 matrix and t is the translational component of the affine transformation. For the rigid case, A is an orthogonal matrix.

The main difficulty in finding an affine transformation that minimizes the error function above is the unknown correspondence. That is, if the correspondence π between points in $\mathcal{P}$ and $\mathcal{Q}$ are known, the following optimization problem

$$E(A, t) = \sum_{i=1}^{k} \|q_{\pi(i)} - p_i\|^2. \tag{2}$$

can be easily solved for the affine transformation (A, t) by solving a system of linear equations. For rigid case with the orthogonality constraint, a slightly more complicated linear-algebraic result will allow us to compute the optimal orthogonal transformation in closed form. However, without knowing the correspondence, any general approach for solving the registration problem posed above invariably requires either some continuous minimization or a discrete variant of it, such as iterative closest point (ICP)[1]. As is well-known, local minimums are usually difficult to avoid and more importantly, it is generally not clear *a priori* that the optimization will indeed converge to the true solution in the ideal case when the absolute global minimum of $\mathcal{E}(A, t)$ of zero can be reached, i.e., noiseless data points.

In this paper, we propose an affine registration algorithm that avoids using optimization and it will guarantee to produce the exact result when the global minimum of $\mathcal{E}$ reaches its lowest possible value of zero. The algorithm first reduces the general affine case to that of the orthogonal case. In the later case, we use the geometry of the complex numbers to explicitly produce a closed-form formula for computing the unknown rotation. Analogous to the interpretation of 3D rotations as unit quaternions [7], 2D rotations can also be interpret as multiplications by unit complex numbers. The main difference between these two interpretations is that the multiplication for complex numbers is commutative, while it is not for quaternions. This, of course, corresponds to the fact that the special orthogonal group $\mathbf{SO}(2)$ is commutative while $\mathbf{SO}(3)$ is not. Treating points in $\mathcal{P}$ and $\mathcal{Q}$ as complex numbers, we can compute polynomials $\mathbf{P}, \mathbf{Q}$ whose roots are the complex numbers in $\mathcal{P}$ and $\mathcal{Q}$, respectively. The relation that the two point sets are related by a 2D orthogonal transformation translates immediately to the fact that the coefficients of the two polynomials are related through powers of some unit complex number. Therefore, by examining the coefficients $\mathbf{P}, \mathbf{Q}$, we are able to recover this unit complex number and hence the rotation. Algebraically, we can compute the rotation in $\mathbb{R}^2$ without knowing the correspondence is a consequence of the fact the $\mathbb{R}^2$ can be equipped with a field structure (complex numbers), and there is no such structure defined on other $\mathbb{R}^n$ with $n \geq 3$ [8].

The main contributions of this paper are

1. A novel affine registration algorithm that is guaranteed to recover the exact affine or rigid transformation in the absence of noise.

2. The algorithm does not require optimization. Furthermore, representing 2D points as complex numbers, the proposed algorithm is easy to implement.

3. We show experimentally that, with minor enhancements, the proposed algorithm performs well also with noisy data.

2 Previous Work

Affine registration is a classical problem in computer vision and the literature on this is quite substantial. It is beyond the scope of this paper to provide even a brief survey on this subject. However, most of the published algorithms require optimization when the correspondences are not known. For example, the successful algorithm of [4] formulates the registration problem using a relaxation scheme with a doubly stochastic matrix in place of the permutation matrix. Spectral algorithms [2][9][10] make up the most important class of registration algorithms that do not require optimization. However, the spectral algorithms require the eigen-structure of some symmetric matrix to be rich enough to provide discriminating features for computing correspondences. To the best of our knowledge, spectral algorithms do not guarantee convergence to the exact solution for every pair of point sets $\mathcal{P}, \mathcal{Q}$. Furthermore, they require computing the eigenvectors of a $k \times k$ matrix, where k is the number of points. This step may be expensive if k is large. In contrast, the algorithm we will describe below will recover the exact solution for every pair of point sets $\mathcal{P}, \mathcal{Q}$ (when the data is noiseless) without optimization or other computationally-expensive steps such as computing eigenvectors.

3 Matching Algorithm

To set the stage for the following discussions, we let $\mathcal{P} = \{p_1, \cdots, p_k\}$ and $\mathcal{Q} = \{q_1, \cdots, q_k\}$ denote two collection of points in $\mathbb{R}^2$. The sizes of the the two point sets are assumed to be the same. We assume that there exists an unknown permutation π on k elements such that

$$q_{\pi(i)} = A\, p_i + t, \tag{3}$$

for some unknown 2×2 nonsingular matrix A and vector t. In the following, we present an algorithm for recovering A, t and the correspondence π given the point sets $\mathcal{P}, \mathcal{Q}$. The translational component t can be disregarded immediately by observing that if m_p and m_q are the centers of $\mathcal{P}$ and $\mathcal{Q}$, respectively, Equation 3 implies that

$$q_{\pi(i)} - m_q = A\,(p_i - m_p), \text{ and } t = m_q - A m_p.$$

That is, we can estimate A using the centered points $q_i - m_q, p_i - m_p$ and the translational component can be recovered once A is determined using the formula above. We will use the same notations $\mathcal{P}$ and $\mathcal{Q}$ to denote the centered point sets.

3.1 Orthogonal Reduction

The centering above gets rid of the translational component. A more sophisticated coordinates transform will allow us to reduce the problem even further: from the four-dimensional problem of determining A (A is 2×2) to that of an one-dimensional problem of determining a rotation.

Let $\mathbf{S_P}$ and $\mathbf{S_Q}$ denote the covariance matrices computed from points in $\mathcal{P}$ and $\mathcal{Q}$, respectively:

$$\mathbf{S_P} = \sum_{i=1}^{k} p_i p_i^t, \qquad \mathbf{S_Q} = \sum_{i=1}^{k} q_i q_i^t.$$

We can assume that the two matrices are positive-definite. For otherwise, each point set belongs to a line and the problem is trivial. We make the following two coordinates transforms using the inverse square-roots of the covariance matrices:

$$p_i \rightarrow \mathbf{S}_{\mathcal{P}}^{-\frac{1}{2}} p_i, \qquad q_i \rightarrow \mathbf{S}_{\mathcal{Q}}^{-\frac{1}{2}} q_i. \tag{4}$$

We will use the same notations to denote the transformed points and point sets. If the original point sets are related by $\mathbf{A}$, the transformed point sets are then related by the linear transformation $\bar{\mathbf{A}} = \mathbf{S}_{\mathcal{Q}}^{-\frac{1}{2}} \mathbf{A} \mathbf{S}_{\mathcal{P}}^{\frac{1}{2}}$. By making such coordinates transforms, the new matrix $\bar{\mathbf{A}}$ can be easily shown to be orthogonal:

Proposition 3.1 *Let $\mathcal{P}$ and $\mathcal{Q}$ denote two point sets (of size k) in $\mathbb{R}^2$, and they are related by an unknown nonsingular matrix $\mathbf{A}$. Assume that their covariance matrices $\mathbf{S}_\mathcal{P}$ and $\mathbf{S}_\mathcal{Q}$ are positive-definite. Then, the transformed point sets (using Equation 4) are related by an orthogonal matrix $\bar{\mathbf{A}}$.*

The proof follows easily from the facts that 1) the covariance matrices $\mathbf{S}_\mathcal{P}$ and $\mathbf{S}_\mathbf{Q}$ are now identity matrices for the transformed point sets, and 2) $\mathbf{S}_\mathbf{Q} = \bar{\mathbf{A}}\mathbf{S}_\mathcal{P}\bar{\mathbf{A}}^t$. They together imply that the rows of $\bar{\mathbf{A}}$ must be orthonormal because $\mathbf{I} = \mathbf{A}\mathbf{A}^t$.

3.2 Orthogonal Case

Now the point sets $\mathcal{P}, \mathcal{Q}$ are related by an orthogonal matrix $\mathbf{R} = \bar{\mathbf{A}}$. Spectral methods such as [2][10] can in principle recover the rotation $\mathbf{R}$ without optimization *provided* that the point sets $\mathcal{P}, \mathcal{Q}$ do not have too much internal symmetry so that the eigenvectors of their proximity matrices (or matrices derived from proximity matrices) are rich enough to supply discriminating features for computing correspondences. Below we present a much simpler algorithm for computing the rotation in R^2 that will recover the correct rotation without optimization for all point sets $\mathcal{P}, \mathcal{Q}$.

The main idea is to use the complex numbers to represent points in $\mathbb{R}^2$, and each point in $\mathcal{P}$ and $\mathcal{Q}$ can be naturally considered as a complex number. A rotation (with determinant 1) in $\mathbb{R}^2$ then corresponds to the multiplication by a complex number with modulus one, i.e., a complex number of the form

$$z = e^{i\theta} = \cos(\theta) + i\sin(\theta),$$

for some real number θ. As is well-known, the orthogonal group $\mathbf{O}(2)$ has two components, with determinant ± 1. The usual reflection across the real axis is given by the action of conjugation: $z \rightarrow \bar{z}$. Therefore, every element $r \in \mathbf{O}(2)$ can be represented as either a multiplication by a unit complex number or multiplication by a unit complex number followed by taking the conjugate:

$$z \rightarrow e^{i\theta} z \quad \text{or} \quad z \rightarrow e^{-i\theta}\bar{z}.$$

Therefore, to determine the unknown rotation $g \in \mathbf{O}(2)$ between $\mathcal{P}$ and $\mathcal{Q}$, we can try to find an element $g \in \mathbf{SO}(2)$ between $\mathcal{P}$ and $\mathcal{Q}$ or between $\mathcal{P}$ and $\bar{\mathcal{Q}}$, conjugates of points in $\mathcal{Q}$.

We define two complex polynomials of degree k:

$$\mathbf{P}(z) = \prod_{i=1}^{k}(z - p_i), \qquad \mathbf{Q}(z) = \prod_{i=1}^{k}(z - q_i). \quad (5)$$

The condition that $\mathcal{P}$ and $\mathcal{Q}$ are related by a rotation implies that there exists a real number θ, such that the two polynomials $\mathbf{P}(z), \mathbf{Q}(e^{i\theta}z)$ are related by an unknown multiplicative constant λ: $\mathbf{Q}(e^{i\theta}z) = \lambda\mathbf{P}(z)$. This follows from the fundamental theorem of algebra because the two polynomials have the same degree and the same set of roots. This implies that they must differ only by a multiplicative constant.

The polynomials $\mathbf{P}(z), \mathbf{Q}(z)$ are computable directly from the point sets. Recall that the elementary symmetric polynomials in k indeterminants are the k linearly independent polynomials below:

$$\begin{aligned}
\Pi_1(x_1, \cdots, x_k) &= \sum_{1 \le i \le k} x_i \\
\Pi_2(x_1, \cdots, x_k) &= \sum_{1 \le i < j \le k} x_i x_j \\
\Pi_3(x_1, \cdots, x_k) &= \sum_{1 \le i < j < l \le k} x_i x_j x_l \\
&\vdots \\
\Pi_k(x_1, \cdots, x_k) &= x_1 \, x_2 \cdots x_k
\end{aligned} \quad (6)$$

The coefficients of $\mathbf{P}(z)$ and $\mathbf{Q}(z)$ are the values of elementary symmetric functions at their roots:

$$a_i = \Pi_i(p_1, \cdots, p_k), \qquad \textbf{and} \qquad b_i = \Pi_i(q_1, \cdots, q_k),$$

if $\mathbf{P}(z) = z^k + a_1 z^{k-1} + a_{k-1}z + a_k$ and $\mathbf{Q}(z) = z^k + b_1 z^{k-1} + b_{k-1}z + b_k$. Furthermore, we also have

$$\mathbf{Q}(e^{i\theta}z) = e^{ik\theta}z^k + b_1 e^{i(k-1)\theta}z^{k-1} + \cdots + e^{i\theta}b_{k-1}z + b_k.$$

Since $\mathbf{Q}(e^{i\theta}z) = \lambda\mathbf{P}(z)$, by matching the coefficients, we have $\lambda = e^{ik\theta}$, and

$$e^{il\theta}b_{k-l} = \lambda a_{k-l}, \longrightarrow e^{i(k-l)\theta} = b_{k-l}/a_{k-l}$$

for $1 \le l \le k-1$. That is, θ is one of the roots of the equation

$$z^d = b_d/a_d$$

for all $0 \le d \le k$, provided that $a_d \neq 0$. The d roots of the above equation are all unit complex numbers and they can be determined quickly by taking the logarithm of b_d/a_d for d such that $a_d \neq 0$: If the complex number $b_d/a_d = x + iy$ for some real numbers x and y,

$$\theta = \frac{1}{d}\tan^{-1}(\frac{y}{x}) + \frac{2n\pi}{d}$$

for $n = 0, 1, 2, \cdots, d-1$. This results in d choices of θ. Some choices of θ may not correspond to the desired rotation. However, every rotation that can match the two point sets must correspond to one of the d choices. Therefore, to determine the rotation, we can take the first nonzero coefficient a_d of $\mathbf{P}(z)$, compute the quotient b_d/a_d, and determine which of the d choices are indeed the desired rotation.

For instance, if $d = 1$, we have $e^{i\theta} = b_1/a_1$, and there is only one θ satisfying this equation. Geometrically, this is easy to see because the coefficients a_1, b_1 are simply the sums of the points in $\mathcal{P}$ and $\mathcal{Q}$, respectively. The desired rotation must then take the center of $\mathcal{P}$ to that of $\mathcal{Q}$. Therefore, if $b_1 \neq 0$, the rotation is determined by the quotient (as complex numbers) of the two centers, a_1/k and b_1/k.

In general, due to the internal symmetry of the point sets, some coefficients a_i (or b_i) will be zero. However, this method will fail only when all coefficients are zero. This is not possible because it will imply that the polynomial $\mathbf{P}(z) = z^k$ and it has a multiple root at 0, which is not how $\mathbf{P}$ is defined.

Recall that the point sets $\mathcal{P}, \mathcal{Q}$ have been centered and with identity matrix as their covariance matrices. A simple algebra will show that

Proposition 3.2 *Let $\mathcal{P} = \{p_1, \cdots, p_k\}$ be a point set in $\mathbb{R}^2$ such that its center is the origin and its covariance matrix is the identity. If $\mathbf{P}(z) = z^k + a_1 z^{k-1} + a_2 z^{k-2} + \cdots + a_k$ is its associated polynomial as defined above, then*

$$a_1 = a_2 = 0.$$

We omit the proof here since it is straightforward. However, it is easy to see why $a_1 = 0$: $a_1 = -(p_1 + p_2 + \cdots + p_k) = 0$ because the center of $\mathcal{P}$ is the origin. With a little more work, one can show that a_2 vanishes as well. Therefore, we need to look for $a_d \neq 0$ with $d \geq 3$. Figure 1 summarizes the proposed 2D affine registration algorithm.

4 Experiments

In this section, we present experimental results for the 2D affine registration algorithm described above. We experimented with both real and synthetic data, and our aim is to provide convincing empirical validations on the accuracy as well as robustness of the proposed method.

4.1 Experiments with Synthetic Data

In these experiments, we randomly generate point sets in $\mathbb{R}^2$ and study the accuracy of the proposed algorithm on these point sets with various degrees of added noise. More specifically, we randomly generate the point set $\mathcal{P} = \{p_1, \cdots, p_{400}\}$ of 400 points in the square domain $-2 \leq x \leq 2, -2 \leq y \leq 2$. The nonsingular matrix A and translational component t are also randomly generated such that their components are required to be in between -2 and 2. The transformed point set $\mathcal{Q} = \{q_1, \cdots, q_{400}\}$ is given by:

$$q_i = A(x_i + n_i^{\delta}) + t,$$

where n_i^{δ} is a randomly generated noise vector such that its two components are independently generated. We experimented with two types of noises: uniform random noise

Given two point sets $\mathcal{P} = \{p_1, \cdots, p_k\}, \mathcal{Q} = \{q_1, \cdots, q_k\}$ in $\mathbb{R}^2$. Determine the affine transformation $\mathbf{A} = (A, t)$ and correspondence π such that

$$q_{\pi(i)} = Ap_i + t$$

for $1 \leq i \leq k$. Assume that A is nonsingular and with positive determinant. Otherwise, we can multiply each point in $\mathcal{Q}$ by the matrix

$$\omega = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

and work with the transformed point set $\mathcal{Q}' = \{\omega q_1, \cdots, \omega q_k\}$.

1. **Center Point Sets**
   Let m_p, m_q denote the centers (of mass) of $\mathcal{P}, \mathcal{Q}$, respectively. Center the two point sets:
   $$p_i \rightarrow p_i - m_p, \qquad q_i \rightarrow q_i - m_q.$$
2. **Orthogonal Reduction**
   Denote the covariance matrices of $\mathcal{P}, \mathcal{Q}$ by $\mathbf{S}_{\mathcal{P}}$ and $\mathbf{S}_{\mathcal{Q}}$, respectively. The following coordinates transforms
   $$p_i \rightarrow \mathbf{S}_{\mathcal{P}}^{-\frac{1}{2}} p_i, \qquad q_i \rightarrow \mathbf{S}_{\mathcal{Q}}^{-\frac{1}{2}} q_i.$$
   make both $\mathcal{P}$ and $\mathcal{Q}$ zero mean and unit variance. The transformed point sets $\mathcal{P}$ and $\mathcal{Q}$ are now related by an unknown rotation $\mathbf{R}$.
3. **Determine Rotation and Correspondence**
   Consider each p_i and q_i as a complex number. Find the smallest integer $n \geq 3$ such that
   $$a_n = \Pi_n(p_1, \cdots, p_k) \neq 0,$$
   $$b_n = \Pi_n(q_1, \cdots, q_k) \neq 0,$$
   where Π_n is the n-th elementary symmetric polynomial. Solve the polynomial equation $z^n = b_n/a_n$. The n roots of the polynomial will contain the desired rotation $\mathbf{R}$, and the correspondence π can be determined by
   $$\pi(i) = \arg \min_{1 \leq l \leq k} \|q_l - \mathbf{R}p_i\|^2.$$
4. **Recover Affine Transformation**
   The nonsingular matrix A and the translational component t are given by
   $$A = \mathbf{S}_{\mathcal{Q}}^{\frac{1}{2}} \mathbf{R} \mathbf{S}_{\mathcal{P}}^{-\frac{1}{2}}, \qquad t = m_q - Am_p.$$

Figure 1. 2D Affine Registration Algorithm

(within $\pm\delta\%$ of the true values (x_i)) and $d\%$ Gaussian random noise (with standard deviation $\delta\%$). We experimented with six different values of δ, $\delta = 0, 2, 4, 8, 10$ and 15.

The errors that we study for each noise setting are the followings. Let (AA, tt) denote the output of the proposed algorithm on $\mathcal{P}$ and $\mathcal{Q}$. We compute the L^2-difference (Frobenius norm) between the exact affine transformation and our estimated result, $\|AA - A\|_2, \|tt - t\|_2$. We also report the relative error $\frac{\|AA-A\|}{\|A\|}$. To compare the quality of the result, we also compute the least-squares errors for both (A, t) and (AA, tt) using Equation 1. For each noise setting, we ran 1000 independent trials (different $\mathcal{P}$, $\mathcal{Q}$ and (A, t) for each trial), and the mean errors obtained from these trials are tabulated in Tables 1

We have implemented the algorithm using MATLAB without any optimization. Using the MATLAB routine 'poly', we can evaluate all elementary symmetric functions Π_n efficiently. The entire implementation contains roughly thirty lines of MATLAB code. The algorithm runs quite efficiently and for each trial, it takes about five to seven seconds to finish on a DELL notebook with a 1.5 GHZ processor.

4.2 Experiments with Real Images

In the subsection, we show some results of applying the proposed algorithm to image registration. In the first experiment, we manually locate twenty feature points on one face image as shown in Figure 2. The face image is taken from the Yale Face Database B, which has been popular in validating face recognition algorithms. It is then rotated by $15^\circ, 30^\circ, 60^\circ$ and 90° to generated four rotated images. We manually locate twenty feature points in each rotated image, which roughly correspond to the twenty points in the original image. We apply the proposed 2D affine registration algorithm to these point sets, and the results are shown in Figure 2. The results show the proposed method is very accurate on estimating the rigid transformations.

In the second experiment, we use images of real-world objects taken under different view points. It is well known that for a small view change, the image difference can be approximated by an affine transformation [6]. Therefore, we apply the proposed algorithm to matching feature points in two images of the same object taken under different viewpoints. We use the COIL datasets from Columbia, which are popular datasets for validating object recognition algorithms. For each of the two objects shown in Figure 3, we select a reference image on which we manually locate twenty feature points. Two other images that were taken 15° and 30° apart from the reference image are also selected for the experiments and as before, we manually locate twenty points on these images which are roughly in correspondence with the feature points on the reference image. The matching results are shown in Figure 3. For piecewise planar surface such as the ANACIN box, it is not surprising that the affine transformation works well in matching these feature points [6]. For the porcelain cat which is not piecewise planar, the proposed algorithm performs surprisingly well for a pair of images with a substantial difference (30°) in viewpoint.

5. Summary and Conclusions

We have proposed a novel affine registration algorithm for matching 2D feature points related by an unknown affine transformation. Given two sets of points in $\mathbb{R}^2$, the algorithm will recover both the unknown affine transformation as well as the correspondence. The proposed algorithm does not require any optimization and therefore, it does not suffer from the usual problem of local minimums. In particular, the algorithm will recover the exact affine transformation and correspondence for data points without noise. Furthermore, the algebraic and geometric motivations behind the proposed algorithm are both clear and transparent. We have also provided experimental results with real and synthetic data demonstrating that the proposed algorithm performs well on noisy data as well.

References

[1] P.J. Besl and N.D. McKay, "A method for registration of 3d shapes", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2), pp. 239-266, 1992.

[2] M. Carcassoni, E. Hancock, "Spectral correspondence for point pattern matching", *Pattern Recognition*, volume 36, pp. 193–204, 2003.

[3] H. Chui and A. Rangarajan, "A new algorithm for non-rigid point matching," *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, volume 2, pp. 44-51, 2000.

[4] S. Gold, A. Rangarajan, C. Lu, S. Pappu and E. Mjolsness, "New algorithms for 2D and 3D point matching: pose estimation and correspondence," *Pattern Recognition* 31(8), pp. 1019-1031, 1998.

[5] G. Golub and C.F. VanLoan, "Matrix Computations", North Oxford Academic, 1996.

[6] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003.

[7] B. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, Volume 4, Issue 4, April 1987, pp.629-642.

[8] T.W. Hungerford, *Algebra*, Springer, 2003.

[9] G. Scott and C. Lonquiet-Higgins, "An algorithm for associating the features of two images," *Proc. of Royal Society of London*, B255, pp. 21-26, 1991.

[10] L. Shapiro and J. Brady, "Feature-based correspondence-an eigenvector approach", *Image Vision Comput.* 10, pp. 283-288, 1992.

Table 1. Experiments with Uniform Noise. For each noise setting, we ran 1000 independent trials. The mean errors are listed with standard deviations shown in parenthesis.

Noise →	0%	2%	4%	8%	10%	15%
Error ↓						
$\|A - AA\|_2$	0 (0)	0.005(0.06)	0.01(0.14)	0.06(0.24)	0.085(0.28)	0.06(0.24)
$\frac{\|A-AA\|_2}{\|A\|_2}$	0(0)	0.003(0.05)	0.01(0.1)	0.04 (0.18)	0.06(0.21)	0.12(0.3)
$\|t - tt\|_2$	0(0)	0 (0)	0.002(0.005)	0.0053(0.0150)	0.06(0.016)	0.0112(0.22)
$\mathcal{E}(A, t)$	0(0)	0.12(0.026)	0.4329 (0.08)	1.2(0.19)	1.6(0.25)	2.2 (0.33)
$\mathcal{E}(AA, tt)$	0 (0)	0.14(0.2)	0.51 (0.3)	1.4 (0.51)	1.8(0.65)	2.5 (0.6)

Table 2. Experiments with Gaussian Noise. For each noise setting, we ran 1000 independent trials. The mean errors are listed with standard deviations shown in parenthesis.

Noise →	0%	2%	4%	8%	10%	15%
Error ↓						
$\|A - AA\|_2$	0 (0)	0.01(0.12)	0.04(0.21)	0.16(0.4)	0.17(0.4)	0.4(0.7)
$\frac{\|A-AA\|_2}{\|A\|_2}$	0(0)	0.01(0.09)	0.11(0.3)	0.04 (0.18)	0.13(0.3)	0.29(0.5)
$\|t - tt\|_2$	0(0)	0.001 (0.008)	0.01(0.02)	0.0053(0.0150)	0.01(0.02)	0.02(0.04)
$\mathcal{E}(A, t)$	0(0)	0.32(0.06)	0.9 (0.15)	1.8(0.3)	2.2(0.38)	2.78 (0.44)
$\mathcal{E}(AA, tt)$	0 (0)	0.36(0.32)	1.1 (0.6)	2.3 (0.68)	2.6(0.7)	3.19 (0.7)

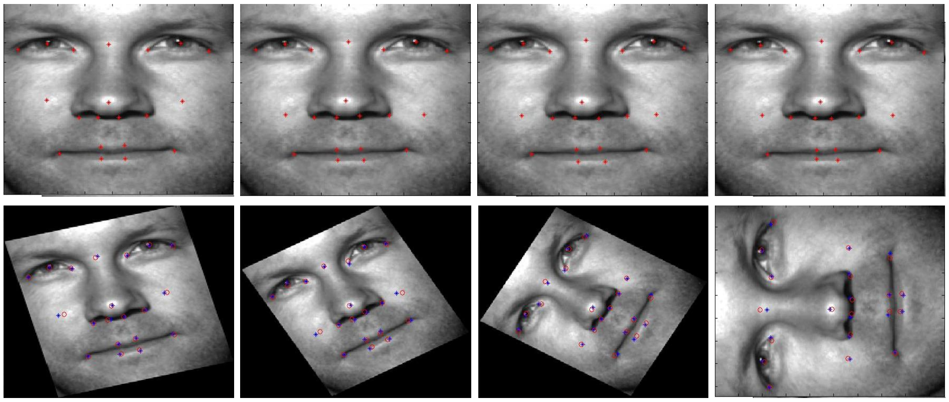

Figure 2. Face Images The top row is the image of one individual in the Yale Face Database B. The images in the bottom are the rotated images of the images above it with degrees $15°, 30°, 60°, 90°$, respectively. We manually locate twenty pairs of roughly corresponding feature points on each image and its rotated image. These are marked with red and blue asterisks on the original and rotated images. The (rigid) registration results are plotted on the rotated images using red circles.

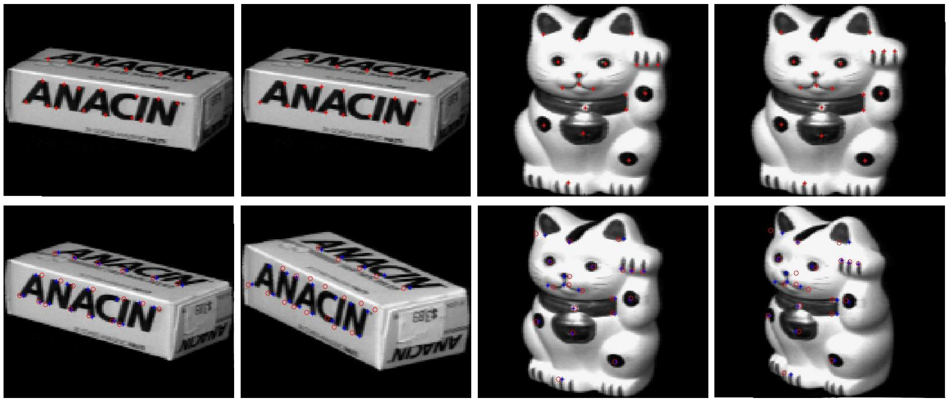

Figure 3. COIL Images The reference images are in the top row. The second row contains images of the same object taken under different viewpoints. The viewpoint differences between vertical pairs of images are $15°, 30°$, respectively. We manually locate twenty feature points on the reference images and the corresponding feature points on the other two images. They are given as red and blue asterisks, respectively. The registration results are marked with the red circles.