

On affine registration of planar point sets using complex numbers

Jeffrey Ho^a, Ming-Hsuan Yang^{b,*}

^a Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32607, United States

^b Electrical Engineering and Computer Science, University of California, Merced, CA 95344, United States

ARTICLE INFO

Article history:

Received 18 June 2009

Accepted 22 July 2010

Available online 30 July 2010

Keywords:

Affine registration

Rigid registration

Point matching

Complex numbers

ABSTRACT

We propose a novel algorithm for affine registration of 2D point sets. The main idea is to treat the 2D points as complex numbers and from each point set, a polynomial with complex coefficients can be computed whose roots are the points in the given point set. The two-step algorithm first reduces the affine registration problem to a rigid registration problem, and the unknown rotation is then computed using the coefficients of these polynomials. The algorithm is entirely algebraic with clear underlying geometric motivation. The implementation is straightforward and it takes less than a second to compute the affine transformation for point sets containing hundreds of points. We validate the algorithm on a variety of synthetic 2D point sets as well as point sets extracted from real-world images.

Published by Elsevier Inc.

1. Introduction

Matching 2D points has been an important and classical problem in computer vision. The problem can be formulated in a variety of ways depending on the applications and their allowable deformations. For instance, affine and rigid registrations have been studied already a while ago, e.g. [1,2], and their applications in computer vision range from shape analysis [3] to medical imaging analysis [4]. Many other applications, particularly those related to medical imaging, require nonrigid registration and recent research activities have focused quite extensively on this subject, e.g. [5–8]. In this paper, we study the more classical problem of matching two point sets in \mathbb{R}^2 related by a rigid (orthogonal) or affine transformation. This problem is still important and relevant today because affine registration is often the crucial first step in many nonrigid registration algorithms as the nonrigid deformation in these algorithms are usually estimated only after a global corrective (affine) transformation has been found to roughly align the point sets. Therefore, an efficient and robust affine registration algorithm could potentially improve significantly the performance of many nonrigid registration algorithms.

Let $\mathcal{P} = \{p_1, \dots, p_k\}$ and $\mathcal{Q} = \{q_1, \dots, q_k\}$ denote two collections of points in \mathbb{R}^2 related by some unknown transformation (rigid or affine) (\mathbf{A}, \mathbf{t}) :

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

where \mathbf{A} , \mathbf{t} are the linear and translational components, respectively. For an affine transformation, \mathbf{A} is represented as a nonsingular

linear 2×2 matrix and for a rigid transformation, it is represented by a 2×2 rotation matrix. The function $\pi: \{1, \dots, k\} \rightarrow \{1, \dots, k\}$ between the two sets of indices gives the correspondence between the points in \mathcal{P} and \mathcal{Q} as the point p_i corresponds to the point $q_{\pi(i)}$. For the moment, we assume that the two point sets contain equal number of points.¹

To formulate an objective function for the affine registration problem for the two point sets \mathcal{P} , \mathcal{Q} , we start with the matching cost function

$$\mathcal{E}_\pi(\mathbf{A}, \mathbf{t}) = \sum_{i=1}^k \|q_{\pi(i)} - \mathbf{A}p_i - \mathbf{t}\|^2, \quad (1)$$

for a fixed correspondence π . Clearly, we have $\mathcal{E}_\pi \geq 0$. Let Π denote a set of correspondences between points in \mathcal{P} , \mathcal{Q} , we define the desired affine transformation to be

$$(\mathbf{A}, \mathbf{t}) = \arg \min_{\mathbf{A}, \mathbf{t}, \pi \in \Pi} \mathcal{E}_\pi(\mathbf{A}, \mathbf{t}). \quad (2)$$

For us, the set Π will be the set of all bijections between the two index sets as we assume the two point sets contain equal number of points. In the noiseless case, there exists at least one correspondence π and (\mathbf{A}, \mathbf{t}) such that $\mathcal{E}_\pi(\mathbf{A}, \mathbf{t})$ attains its smallest possible value of zero. In general, without restriction on the correspondences in Π , the problem posed in (2) cannot be solved because of the degenerate correspondences (e.g., the range of π contains exactly one point).

The main difficulty in finding an affine transformation that minimizes the error function above is the unknown correspondence π .

* Corresponding author.

E-mail addresses: jho@cise.ufl.edu (J. Ho), mhyang@ucmerced.edu (M.-H. Yang).

¹ We will discuss methods for dealing with point sets of different sizes later in this paper.

That is, if the correspondence π between points in \mathcal{P} and \mathcal{Q} are known, the optimization problem

$$(\mathbf{A}, \mathbf{t}) = \arg \min_{\mathbf{A}, \mathbf{t}} \mathcal{E}_{\pi}(\mathbf{A}, \mathbf{t}) = \arg \min_{\mathbf{A}, \mathbf{t}} \sum_{i=1}^k \|q_{\pi(i)} - \mathbf{A}p_i - \mathbf{t}\|^2 \quad (3)$$

can be easily solved 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 [9]. 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), e.g. [10,11]. As is well-known, local minimums are usually difficult to avoid and more importantly, it is generally not clear *a priori* that the algorithm will indeed converge to the true solution even in the noiseless case. The problem with local minimums can be alleviated somewhat using simulated annealing [12]; however, this makes the method very inefficient and impractical for many applications.

In this paper, we propose a novel affine registration algorithm that avoids using optimization, and it will guarantee to produce the exact result when the data points contain no noise. 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 [9],² 2D rotations can also be interpret as multiplications by unit complex numbers. The main difference between these two cases 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 for higher-dimensions, the relevant algebraic structures are the Clifford algebras [13], which are generally not commutative.

Needless to say, the literature on affine registration is immense and extensive. However, in the context of what has been already known about 2D affine registration, our paper has following two specific contributions:

1. A novel affine registration algorithm that does not require continuous optimization and in the absence of noise, it will recover the exact affine transformation. The algorithm is very easy to implement and it is very efficient for medium-size point sets containing hundreds of points.
2. Experimental results that validate the proposed algorithm. Using both synthetic and real 2D point sets, we demonstrate that the proposed algorithm does indeed provide an efficient and robust solution for affine registration problems.

We believe that the proposed algorithm has a great potential in many applications as it can be used either as a stand-alone registration algorithm or as an efficient method for producing good ini-

tializations for registration algorithms that use more elaborated nonlinear optimization, e.g. [12].

2. Related work

Affine registration is a classical problem in computer vision and there is an extensive body of literature on this subject, e.g. [14–22]. It is certainly beyond the scope of this paper to provide a detailed survey on this topic, and here we discuss only the most relevant work for matching 2D point sets. Broadly speaking, the existing algorithms can be categorized into three groups: invariant-based methods, e.g. [23–26], optimization-based methods, e.g. [5,27,10,6,12,21,28,11] and spectral matching methods, e.g. [1,29–31]. For the invariant-based methods, Hu's work on moment invariants [32,33] has been particularly influential. The idea behind this type of approaches is to define numerical quantities constructed from various moments that are invariant under affine transforms. These invariants can be specialized to define affine-invariant features for each point [25,26], and the correspondences can be computed as a global linear or quadratic assignment problem using these local invariant features. However, the affine-invariant features are usually constructed from moments with degree greater than two, and as the invariants typically contain multiple products of these moments, the effective degrees of the invariants are generally larger than four or five, which makes them sensitive to noise. Furthermore, the algebra required to define these invariants can be quite involved, e.g. [33], and computing correspondences from local features almost always requires one more layer of algorithmic component that increases the algorithm's complexity. Therefore, our method compared favorably with these algorithms in terms of simplicity in its underlying theory as well as implementation. Besides moment-based invariants, geometric invariants have also been investigated, e.g. [34,35]. For example, the edges of the point-set's convex hull can serve as important features for matching was first noted in [36] for similarity transforms, and the vertices of the convex hull were later used for affine point-set matching in [37,38]. Affine invariants are constructed using four consecutive vertices of a convex hull at a time. These invariants are then used to estimate a global affine transform between the two point sets. Unfortunately, these local geometric invariants are even more sensitive to noise than the moment invariants as small local perturbations of the points can alter the invariants considerably to render them ineffective.

The Iterative Closest Point (ICP) algorithm by Besl and McKay [10] is perhaps the most widely used technique in point matching. The method iteratively solves for the correspondence and transformation, and it can be applied to both affine and rigid registrations. The correspondence is determined using the nearest neighbor with a fixed transformation, and the transformation is determined by solving a linear system of equations when the correspondence is fixed. This interleaving process of solving correspondence and transformation is repeated until it reaches a local minimum. We note however that there exist no theoretical results that provide some guarantee on the ICP outcome. In particular, even for noiseless data, it is not clear if ICP would in fact converge to the right solution. A more elaborated method appeared in [39], where the registration problem is solved using a relaxation scheme with a doubly stochastic matrix (soft assignment) in place of the permutation matrix (hard assignment). The registration error is minimized with nonlinear optimizer using the Levenberg–Marquardt algorithm (also used in [11]).

Spectral algorithms [1,29,31] make up the most important class of registration algorithms that do not require optimization. The well-known algorithm by [1] solves the registration problem by aligning the (unit) eigenvectors of the covariance matrices of the two point sets. It has been extended and generalized in several dif-

² We note that in [9] the correspondence π is known and in our case, it is unknown.

ferent ways for matching points and shapes in \mathbb{R}^2 [30,40,31]. However, for a spectral algorithm to work, it requires the eigen-structure of some symmetric matrix to be rich enough so as to provide sufficiently discriminating features for computing correspondence. To the best of our knowledge, the spectral algorithms do not guarantee to converge to the exact solution for every pair of point sets \mathcal{P} , \mathcal{Q} . Furthermore, they also require computing the eigenvectors and eigenvalues 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 does not require any computationally-expensive step, and the algorithm will recover the exact affine transformation and correspondence for every pair of point sets \mathcal{P} , \mathcal{Q} when there is no noise in the data.

Finally, there are other algorithms that do not easily fit into the three broad categories described above. For example, a fast 2D clustering-based approach is introduced in [41]. [42] uses a least-squares approach for registering two point sets with equal number of points, and in [43], distances between every pair of points are calculated for each point set, and some heuristics are used to compute local matches between the point sets using these pairwise distances. More involved computational structures such as graphs and search trees can be used to provide more efficient and better methods for determining the correspondences [44,45]. A different approach using statistical methods appears in [46], where the proposed method iteratively computes the registration and correspondence using Procrustes and point-distribution models. [47] proposes a geometric alignment strategy for registering images, and the concept of geometric hashing is used in [48]. Again, compared with our methods, these earlier methods are considerably more involved both in their underlying theory and implementation.

3. Matching algorithm

In this section, we detail the proposed matching algorithm. Our main contribution is the idea of using complex numbers to solve rigid registration. The general affine registration is then solved by reducing it to a corresponding rigid registration. This orthogonal reduction step is based on the idea of normalizing the covariance matrices, and it is a well-known method in computer vision and machine learning. While we do not claim originality for this step, for completeness, we provide all the necessary details in Section 3.1. The remaining sections will be focused on using complex numbers to solve rigid registration.

To fix the notations for the following discussion, we let $\mathcal{P} = \{p_1, \dots, p_k\}$ and $\mathcal{Q} = \{q_1, \dots, q_k\}$ denote two collections of points in \mathbb{R}^2 . The sizes of the two point sets are assumed to be the same for the moment. π , a permutation on k elements, will denote the correspondence between the two point sets such that

$$q_{\pi(i)} = \mathbf{A}p_i + \mathbf{t}, \quad (4)$$

for some 2×2 nonsingular matrix \mathbf{A} and 2×1 vector \mathbf{t} . Our goal is to recover the linear \mathbf{A} and translational \mathbf{t} components of the affine transformation between the two point sets. The translational component \mathbf{t} can be immediately disposed of by observing that if m_p and m_q are the centers of mass for \mathcal{P} and \mathcal{Q} ,

$$m_p = \frac{1}{k} \sum_{i=1}^k p_i, \quad m_q = \frac{1}{k} \sum_{i=1}^k q_i,$$

(4) then implies that

$$q_{\pi(i)} - m_q = \mathbf{A}(p_i - m_p)$$

with

$$\mathbf{t} = m_q - \mathbf{A}m_p.$$

That is, we can estimate \mathbf{A} using the centered points $q_i - m_q$, $p_i - m_p$, and the translational component \mathbf{t} can be recovered once

\mathbf{A} is determined using the equation above. Henceforth, we assume that the points in \mathcal{P} , \mathcal{Q} are centered and

$$q_{\pi(i)} = \mathbf{A}p_i, \quad (5)$$

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 further: from the four-dimensional problem of determining \mathbf{A} to that of an one-dimensional problem of determining a rotation. As will be clear soon, the combined step here is to normalize the two point sets so that they have zero mean and unit covariance, a normalization step often employed in computer vision and machine learning for processing training data.

The basic idea behind the orthogonal reduction step is to determine the unknown matrix \mathbf{A} up to a rotation by requiring it to preserve the two covariance matrices

$$\mathbf{S}_P = \frac{1}{k} \sum_{i=1}^k p_i p_i^t, \quad \mathbf{S}_Q = \frac{1}{k} \sum_{i=1}^k q_i q_i^t.$$

If (5) is satisfied, then the two covariance matrices are related as

$$\mathbf{S}_Q = \mathbf{A}^t \mathbf{S}_P \mathbf{A}. \quad (6)$$

Therefore, \mathbf{A} is a solution of (6). If we assume that the two covariance matrices are positive-definite (otherwise, the point sets lie on lines and the problem reduces to matching points on \mathbb{R}^1), (6) can always be solved up to an unknown rotation. This is a special case of a more general theorem [49]

Theorem 3.1. *For any pair \mathbf{S} , \mathbf{T} of $n \times n$ symmetric positive-definite matrices, there exists a nonsingular $n \times n$ matrix \mathbf{A} such that*

$$\mathbf{S} = \mathbf{A}^t \mathbf{T} \mathbf{A} \quad (7)$$

Furthermore, any matrix of the form

$$\mathbf{A} = \mathbf{T}^{-\frac{1}{2}} \mathbf{R} \mathbf{S}^{\frac{1}{2}}$$

with \mathbf{R} an $n \times n$ rotation matrix satisfies (7).

It is easy to verify that such \mathbf{A} does satisfy (7) with $\mathbf{S}^{\frac{1}{2}}$, $\mathbf{T}^{-\frac{1}{2}}$ denoting the square-root and inverse square-root of \mathbf{S} , \mathbf{T} , respectively. Using this theorem, we can define coordinates transforms using the inverse square-roots of the covariance matrices

$$p_i \rightarrow \mathbf{S}_P^{-\frac{1}{2}} p_i, \quad q_i \rightarrow \mathbf{S}_Q^{-\frac{1}{2}} q_i. \quad (8)$$

If the original point sets are related by \mathbf{A} , the transformed point sets are then related by the linear transformation $\mathbf{R} = \mathbf{S}_Q^{-\frac{1}{2}} \mathbf{A} \mathbf{S}_P^{\frac{1}{2}}$. The theorem above immediately implies that \mathbf{R} is a rotation matrix, and we have

Proposition 3.2. *Let \mathcal{P} and \mathcal{Q} denote two point sets in \mathbb{R}^2 , and they are related by an unknown nonsingular matrix \mathbf{A} as in (5). Assume that their covariance matrices \mathbf{S}_P and \mathbf{S}_Q are positive-definite. Then, the transformed point sets (using (8)) are related by an orthogonal matrix \mathbf{R} .*

The proof follows easily from the theorem above, or one can prove directly that (1) the covariance matrices \mathbf{S}_P and \mathbf{S}_Q are now the identity matrix \mathbf{I} for the transformed point sets, and (2) $\mathbf{S}_Q = \mathbf{R}^t \mathbf{S}_P \mathbf{R}$ becomes $\mathbf{I} = \mathbf{R}^t \mathbf{R}$. Two examples of the normalization are shown in Fig. 1.

3.2. Rigid registration

Again, we will use the same notations to denote the transformed points and point sets (8), and the point sets \mathcal{P} , \mathcal{Q} are

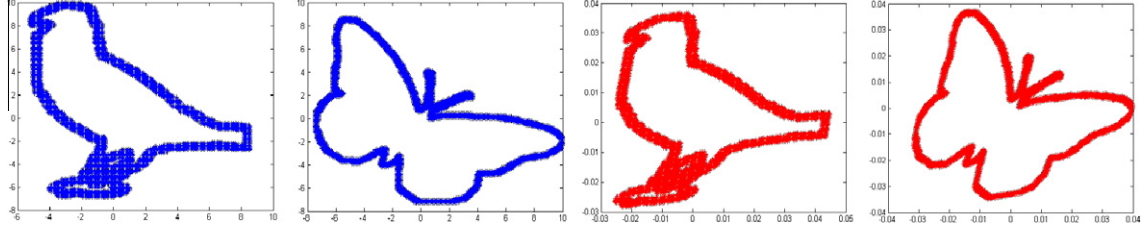


Fig. 1. Orthogonal reduction. Two examples of the coordinates transform (8) that normalizes the covariance matrices. The first two images from the left give the original point sets with 2927 and 2866 points, respectively. The normalized point sets are shown in the last two images. Notice the scale change as well as some amount of shearing in the normalized point sets. The covariance matrices of the normalized point sets are the identity matrix.

now related by an orthogonal matrix \mathbf{R} , a rigid registration problem. Spectral methods such as [29,31] 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. For example, [29] solves the rigid registration problem by matching the eigenvectors of the two covariance matrices. Unfortunately, this method is not applicable here because the two covariance matrices are now the identity matrix, and there is no way to match the eigenvectors using different eigenvalues as in [29]. On the other hand, [31] requires the computation of a “proximity matrix”, and it uses the eigenvectors of this matrix as features for computing correspondences. Therefore, for this method to work, the eigen-structure of the proximity matrix must be rich enough to supply sufficient discriminative features for computing correspondences. Here, we present a much simpler algorithm for solving the rigid registration using the geometry of complex numbers.

We will identify every point in \mathbb{R}^2 with a complex number in the usual manner: $(x, y) \rightarrow x + iy$, and each point in \mathcal{P} and \mathcal{Q} is now considered as a complex number. A rotation (with determinant 1) in \mathbb{R}^2 then corresponds to the multiplication by a complex number with unit modulus, 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 group $\mathbf{O}(2)$ of orthogonal matrices has two connected components: the component $\mathbf{SO}(2)$ with rotation matrices of determinant 1 and the other component with matrices of determinant -1 . The reflection ω across the real (x -) axis is a rotation matrix with determinant -1

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

and in the complex setting, it is given by the action of conjugation: $z \rightarrow \bar{z}$. With this, every matrix $\mathbf{R} \in \mathbf{O}(2)$ can be represented as either a multiplication by a unit complex number or multiplication by a unit complex number followed by taking the conjugate:

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

for some θ . 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 polynomials of degree k with complex coefficients:

$$\mathbf{P}(z) = \prod_{i=1}^k (z - p_i), \quad \mathbf{Q}(z) = \prod_{i=1}^k (z - q_i). \quad (9)$$

The condition that \mathcal{P} and \mathcal{Q} are related by a (orientation preserving) 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, and it follows that they must differ only by a multiplicative constant.

Let $\mathbf{P}(z) = z^k + a_1 z^{k-1} + \dots + a_{k-1} z + a_k$ and $\mathbf{Q}(z) = z^k + b_1 z^{k-1} + \dots + b_{k-1} z + b_k$. The coefficients a_i , b_i can be computed using elementary symmetric polynomials in k indeterminates: these are the k linearly independent polynomials with degrees $1, 2, \dots, k$:

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

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

$$a_i = \Pi_i(p_1, \dots, p_k), \quad \text{and} \quad b_i = \Pi_i(q_1, \dots, q_k).$$

A quick calculation shows that

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

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

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

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

$$z^d = b_d/a_d$$

for all $1 \leq d \leq k-1$, 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 complex logarithm of b_d/a_d for d such that $a_d \neq 0$: if $b_d/a_d = x + iy$ for some real numbers x and y ,

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

for $n = 0, 1, 2, \dots, d-1$. This results in d choices for 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 .

Recall that the point sets \mathcal{P} , \mathcal{Q} have been centered and their covariance matrices are the identity matrix. A simple algebra will show that the first two coefficients a_1 , b_1 , a_2 , b_2 must vanish:

Proposition 3.3. Let $\mathcal{P} = \{p_1, \dots, p_k\}$ be a point set in \mathbb{R}^2 such that its center is the origin and its covariance matrix is the identity. If $\mathbf{P}(z) = z^k + a_1 z^{k-1} + a_2 z^{k-2} + \dots + a_k$ is its associated polynomial as defined above, then

$$a_1 = a_2 = 0.$$

Proof. Let $p_j = x_j + i y_j$ for $1 \leq j \leq k$. The coefficient a_1 is simply the sum of p_i , which is zero because \mathcal{P} is centered. To show that a_2 is also zero, we first observe that

$$a_2 = \sum_{1 \leq i < j \leq k} (x_i x_j - y_i y_j) + i \sum_{1 \leq i < j \leq k} (x_i y_j + x_j y_i).$$

The condition that \mathcal{P} has zero mean and unit variance implies that

$$x_1 + \dots + x_k = 0, \quad y_1 + \dots + y_k = 0 \quad (10)$$

$$x_1^2 + \dots + x_k^2 = 1, \quad y_1^2 + \dots + y_k^2 = 1 \quad (11)$$

$$x_1 y_1 + x_2 y_2 + \dots + x_k y_k = 0$$

The first two equations above imply that the real part of a_2 is zero because by squaring the first pair of equations and using the second pair of equations, it shows that

$$\sum_{1 \leq i < j \leq k} x_i x_j = \sum_{1 \leq i < j \leq k} y_i y_j.$$

□

Similarly, the first and third equations above together imply that the imaginary part of a_2 vanish as well.

There are no other constraints on \mathcal{P} that will force a_3 to be zero as a_1, a_2 are. However, it can vanish if the point sets contain enough internal symmetry. Nevertheless, our method will fail only when all (non-leading) 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. In general, we expect that coefficient a_3 to be nonzero for most point sets, and we can use a_3, b_3 to recover the rotation matrix and hence the affine transformation. While a_3 (and b_3) is a coefficient in the high-degree polynomial \mathbf{P} , the stability of a_3 against noise is not related to the high-degree polynomial \mathbf{P} as a_3 is the sum of products of three points in \mathcal{P} :

$$a_3 = \sum_{1 \leq i < j < l \leq k} p_i p_j p_l,$$

and its degree with respect to the input points p_i is three. Similarly, the degree of a_4 is four, etc. As the low-degree quantities are generally more robust against noises compared with high-degree quantities, a_i with a small value of $i \geq 3$ should be preferred for recovering the rotation. The full 2D affine registration is summarized in Fig. 2.

For noisy data, the proposed algorithm can be enhanced by an appended step that computes the optimal affine transformation given the correspondence computed by the algorithm. Specifically, let $\mathbf{A} = (A, t)$ denote the computed affine transformation. The correspondence π is determined by finding the nearest neighbor:

$$\pi(i) = \arg \min_{1 \leq j \leq k} \|q_j - A x_i - t\|_2.$$

As mentioned before, the optimal affine transformation with correspondence π (1) can be computed by solving a system of linear equations.

3.3. Point sets of different sizes

It is possible to generalize the method described above for point sets of different size. Without loss of generality, we assume that \mathcal{P} has more points than \mathcal{Q} . The rigid reduction part of the algorithm does not require equal number of points as it only uses the covari-

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

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

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 $\mathcal{Q}' = \{\omega q_1, \dots, \omega q_k\}$.

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:

$$p_i \rightarrow p_i - m_p, \quad q_i \rightarrow q_i - m_q.$$

2. Orthogonal Reduction

Denote the covariance matrices of \mathcal{P}, \mathcal{Q} by \mathbf{S}_P and \mathbf{S}_Q , respectively. The following coordinates transforms

$$p_i \rightarrow \mathbf{S}_P^{-\frac{1}{2}} p_i, \quad q_i \rightarrow \mathbf{S}_Q^{-\frac{1}{2}} q_i.$$

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

$$a_n = \Pi_n(p_1, \dots, p_k) \neq 0,$$

where Π_n is the n -th elementary symmetric polynomial. Let $b_n = \Pi_n(q_1, \dots, q_k)$. 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

$$A = \mathbf{S}_Q^{-\frac{1}{2}} \mathbf{R} \mathbf{S}_P^{-\frac{1}{2}}, \quad t = m_q - A m_p.$$

Fig. 2. 2D affine registration algorithm.

ance matrices of the point sets. However, the rigid registration part that follows does require \mathcal{P}, \mathcal{Q} to have the same number of points. Otherwise, the polynomials \mathbf{P} and \mathbf{Q} would have different degrees and we no longer have $\mathbf{Q}(e^{i\theta} z) = \lambda \mathbf{P}(z)$ for some multiplicative constant λ . Instead, we have $\mathbf{Q}(e^{i\theta} z) | \mathbf{P}(z)$, i.e., $\mathbf{P}(z)$ is divisible by $\mathbf{Q}(e^{i\theta} z)$. This will provide a set of relations between the coefficients of \mathbf{P} and \mathbf{Q} . In principle, these relations can be utilized in a similar way as before to determine the unknown rotation; however, the algebra involved here is considerably more complicated and we will not pursue this approach here.

Instead, we will investigate two different approaches for handling point sets of different size. In both methods, the main idea is to produce modified point sets $\mathcal{P}', \mathcal{Q}'$ from \mathcal{P}, \mathcal{Q} such that $\mathcal{P}', \mathcal{Q}'$ have the same number of points and the registration algorithm can be directly applied to $\mathcal{P}', \mathcal{Q}'$.

3.3.1. Random point deletion

Given the two point sets \mathcal{P}, \mathcal{Q} with different number of points, one quick solution is to modify \mathcal{P} by randomly deleting its points so that the resulting point set \mathcal{P}' has the same number of points as \mathcal{Q} . The two advantages of this method are its easy implementation and the negligible increase in the overall running time. In particular, for point sets that differ no more than 15% in their sizes, our experiments have shown that the registration results are generally satisfactory. However, the main shortcoming of this method is that it does not provide a principled way to reduce the number of points for both point sets. For point sets that differ significantly in their sizes (say more than 20%) or point sets containing too many points, it is desirable to have a principled method that can produce a modified point set from the original

point set with significantly fewer number of points that still retains sufficiently many of its original geometric features. For example, for point sets containing tens of thousands of points, it is desirable to first reduce the number of points to the order of thousands or hundreds of points and register the simplified point sets. Unfortunately, for this type of drastic reductions (typically more than 90% reduction), random point deletion is not particularly effective, and the second method improves on this shortcoming, at the expense of longer running time, using the idea of point resampling.

3.3.2. Point resampling

The main idea here is to treat the points in the point set \mathcal{P} as sample points drawn from an unknown probability density function (PDF). Therefore, if we can estimate the density function, we can sample with respect to it a new set \mathcal{P}' of points of any given size, and this gives us a principled way for obtaining the simplified point sets \mathcal{P}' , \mathcal{Q}' that have the same number of points. This approach can be justified for many practical applications. For example, many affine registration problems involve aligning 2D planar shapes given as point sets (Fig. 3). The aim here is to align the shapes implied by the point sets, and the actual matching of each individual point is only of secondary importance. In this context, the shapes implied by the point sets are represented as probability density functions, and the registration problem turns into a registration problem for two PDFs defined on \mathbb{R}^2 , which can be formulated and solved in various ways. Given a pair of PDFs, $\mathcal{F}_{\mathcal{P}}$, $\mathcal{F}_{\mathcal{Q}}$, the most straightforward method would be to align the two PDFs based on the ℓ_2 -error [50]:

$$(\mathbf{A}, \mathbf{t}) = \arg \max_{\mathbf{A}, \mathbf{t}} \int_{\mathbb{R}^2} \|\mathcal{F}_{\mathcal{Q}}(\mathbf{x}) - \mathcal{F}_{\mathcal{P}}(\mathbf{A}\mathbf{x} + \mathbf{t})\|^2 d\mathbf{x}.$$

Different distance measures for the density functions, such as the various divergences, can be substituted for the ℓ_2 -norm. The optimization problem is typically solved using gradient descent, and as usual, the quality of the solution cannot be guaranteed in general. To make our method directly applicable here, we opt for a simpler solution that registers two new point sets containing the same number of points obtained by sampling the two PDFs.

Density estimation, particularly on \mathbb{R}^2 , is a well-studied subject, encompassing a variety of techniques and mathematical formulations. We choose to fit the point set \mathcal{P} with a Gaussian Mixture Model (GMM) of K components, where K is an integer that can be either user-specified or determined by the required precision for the fitting [51]: the density function $\mathcal{F}_{\mathcal{P}}$ is given in the form

$$\mathcal{F}_{\mathcal{P}}(\mathbf{x}|\Theta) = \sum_{i=1}^K \omega_i \mathbf{N}(\mathbf{x}; \mu_i, \Sigma_i)$$

where

$$\mathbf{N}(\mathbf{x}; \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{N/2} |\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_i)^t \Sigma_i^{-1}(\mathbf{x} - \mu_i)\right)$$

is the 2D normal distribution with mean μ_i and covariance matrix Σ_i , and ω_i the non-negative weights that sum to one. $\Theta = \{\mu_1, \dots, \mu_K, \Sigma_1, \dots, \Sigma_K, \omega_1, \dots, \omega_K\}$ denote the parameters of the Gaussian Mixture Model, and the parameters can be estimated by maximizing the (log) likelihood

$$\mathbf{L}_{\mathcal{P}}(\Theta) = \prod_{i=1}^K \mathcal{F}_{\mathcal{P}}(p_i|\Theta).$$

This can be solved using the EM algorithm, and the details are available in many textbooks (e.g., [51]). Once the density function $\mathcal{F}_{\mathcal{P}}$ has been estimated, the sampling with respect to it is straightforward: first, we sample the component according to the weights $\{\omega_1, \dots, \omega_K\}$. Once the component is given, sampling with respect to $\mathbf{N}(\mathbf{x}; \mu_i, \Sigma_i)$ is straightforward.

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 validation on the accuracy as well as robustness of the proposed method. We have implemented the algorithm in C++ and MATLAB. Both implementations are without optimization, and for point sets containing roughly 500 points, the MATLAB and C++ implementations take, on average, 5–7 s and less than one second to compute the affine transformation on a DELL notebook with a 1.5 GHz processor, respectively. On MATLAB, the polynomials defined in (9) can be computed efficiently using the MATLAB routine `poly`, and the entire MATLAB implementation contains roughly 30 lines of code. We compare the performance of our algorithm with that of the affine variant of the usual ICP algorithm that iteratively computes the transformation and correspondences.

4.1. Experiments with synthetic data

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

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

where n_i^δ is a randomly generated noise vector whose two components are independently generated. We experimented with two types of noise: uniform random noise (within $\pm\delta\%$ of the true value x_i) and Gaussian random noise (with standard deviation $\delta\%$). We experimented with five different values of δ , $\delta = 0, 2, 4, 8$, and 10 .

For each noise setting, the accuracy of our method is measured by computing several error values. Let (A_{est}, t_{est}) denote the output

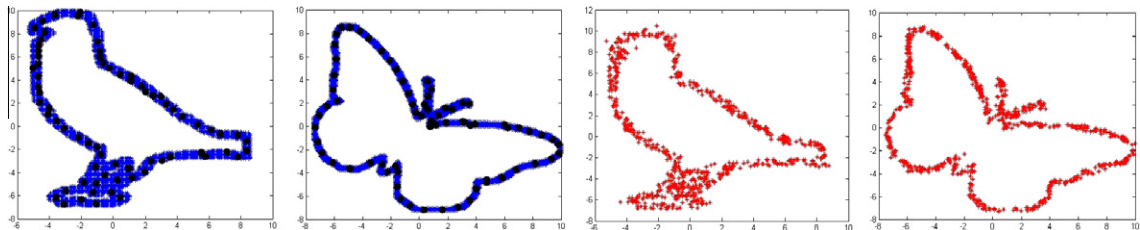


Fig. 3. Point sets of the different sizes. The first two images show the original point sets with 2927 and 2866 points, respectively. A 70-component Gaussian Mixture Model is fitted to each point set and the centers of the mixture components are marked black in the figures. The two new sampled point sets with 700 points are shown in the last two images.

of our algorithm for the input point sets \mathcal{P} and \mathcal{Q} . We compute the ℓ_2 -difference (Frobenius norm) between the exact affine transformation and our estimated result, $\|A_{est} - A\|_2$, $\|t_{est} - t\|_2$, and also the relative error $\frac{\|A_{est} - A\|_2}{\|A\|_2}$. To compare the quality of the registration result, we also compute the errors for both (A, t) and (A_{est}, t_{est}) using (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 using uniform and Gaussian noises are tabulated in Tables 1 and 2, respectively.

The results show that our method, which is entirely algebraic, does tolerate some degrees of input noises. We note that when there is no added noise, our method does indeed recover the correct affine transformation as all the error values are zero. In all the experiments, the relative error $\frac{\|A_{est} - A\|_2}{\|A\|_2}$ of the estimated affine transformation is less than 13% of the true affine transformation A . This shows that the affine transformation A_{est} recovered by our method is indeed quite close to the actual transformation A for noisy point sets.

For comparison, we test the affine variant of the ICP algorithm. This algorithm iteratively computes the transformation and correspondence. Given the transformation, the correspondence is computed using the nearest neighbors: for each $p \in \mathcal{P}$, its corresponding point q in \mathcal{Q} is defined as

$$q = \arg \min_{q \in \mathcal{Q}} \|\bar{q} - Ap - t\|^2.$$

We note that the resulting correspondence can fail to be surjective, and multiple points in \mathcal{P} can be assigned to the same point in \mathcal{Q} . Conversely, given the correspondence, the affine transform is computed by minimizing (1), which can be done efficiently by solving two linear systems for A , t separately. The algorithm is initialized using the identity transform as the initial affine transform, and the results are tabulated in Table 3. The poor performance is not surprisingly since the algorithm is known to depend on the initialization. In particular, the registration accuracies do not generally depend on the amount of added noise, which implies that the effect of poor initializations typically overrides the effect of noise. For most optimization-based methods, a good initialization is usually a prerequisite for the success of the algorithms. Unfortunately, for most applications, good initializations are often unavailable if the algorithm does not assume any prior information.

4.2. Experiments on registering 2D shapes

In this section, we report results of applying our method to registering 2D shapes. The shapes used in our experiments are taken from the MPEG-7 shape database. In this database, there are 70 different shape categories with 20 shapes per category. All shapes in this database are represented as point sets containing roughly between 1000 and 3000 points.³ Using this database, we report two sets of experiments that qualitatively and quantitatively study the accuracy of our method.

In the first set of experiments, two images from the same shape category are randomly chosen for registration. Since no ground truth can be established for this experiment, five registration results are shown in Fig. 4 without any quantification on their error and accuracy. The quality of the registration result is determined visually, and as can be seen from the figure, our automatic registration method does quite well for affine aligning the pairs of shapes belonging to the same shape category. Since the point sets in general do not contain the same number of points, we use the point resampling method described in the previous section by fitting a K -component Gaussian Mixture Model to each point set. In all the experiments, K is taken to be 70 and we sample 700 points

Table 1

Experimental results with uniform noise. For each noise setting, we ran 1000 independent trials. The mean errors are listed with standard deviations shown in parenthesis.

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

Table 2

Experimental results with Gaussian noise. For each noise setting, we ran 1000 independent trials. The mean errors are listed with standard deviations shown in parenthesis.

Noise → Error ↓	0%	2%	4%	8%	10%
$\ A - A_{est}\ _2$	0 (0)	0.01(0.12)	0.04 (0.21)	0.16 (0.4)	0.17 (0.4)
$\frac{\ A - A_{est}\ _2}{\ A\ _2}$	0 (0)	0.01 (0.09)	0.02 (0.2)	0.04 (0.18)	0.13 (0.3)
$\ t - t_{est}\ _2$	0 (0)	0.001 (0.008)	0.01 (0.02)	0.0053 (0.0150)	0.01 (0.02)
$\mathcal{E}(A, t)$	0 (0)	0.32 (0.06)	0.9 (0.15)	1.8 (0.3)	2.2 (0.38)
$\mathcal{E}(A_{est}, t_{est})$	0 (0)	0.36 (0.32)	1.1 (0.6)	2.3 (0.68)	2.6 (0.7)

Table 3

Experimental results using affine ICP with Gaussian noise. For each noise setting, we ran 1000 independent trials. The mean errors are listed with standard deviations shown in parenthesis.

Noise → Error ↓	0%	2%	4%	8%	10%
$\ A - A_{est}\ _2$	0.45 (0.33)	0.51 (0.32)	0.62 (0.45)	0.55 (0.61)	0.54 (0.51)
$\frac{\ A - A_{est}\ _2}{\ A\ _2}$	0.81 (0.45)	0.83 (0.51)	0.79 (0.43)	0.85 (0.61)	0.96 (0.53)
$\ t - t_{est}\ _2$	0.03 (0.06)	0.04 (0.06)	0.05 (0.04)	0.06 (0.06)	0.07 (0.06)
$\mathcal{E}(A, t)$	3.5 (1.4)	3.7 (1.7)	3.9 (1.9)	3.6 (2.1)	4.1 (1.7)
$\mathcal{E}(A_{est}, t_{est})$	3.7 (1.2)	3.6 (1.4)	3.7 (1.2)	4.1 (1.2)	3.9 (1.3)

from each GMM to form the new point set, which is roughly 80% reduction in the number of points. For each pair of shapes, it takes about, on average, several minutes for the algorithm to finish running a MATLAB implementation. Most of computation time is however spent on estimating the two Gaussian Mixture Models, and the actual registration step takes only a few seconds to complete as before.

The second set of experiments provide a quantitative analysis on the accuracy of our method for registering point sets containing different number of points using randomly point deletion. In this experiment, we randomly select five shapes from the MPEG-7 shape database (shown in Fig. 5) and each shape is represented as a point set \mathcal{P} . A percentage of the points are deleted from \mathcal{P} and a randomly generated affine transformation (A, t) is then applied to the resulting point set to form a new point set \mathcal{Q} . The two point sets \mathcal{P} , \mathcal{Q} are then related by a known affine transformation but containing different number of points. Again, let (A_{est}, t_{est}) denote the output of our method for the input point sets \mathcal{P} and \mathcal{Q} , and the error of interest here is the relative error $\frac{\|A_{est} - A\|_2}{\|A\|_2}$ between

³ We thank Anand Rangarajan and Adrian Peter for providing the data.

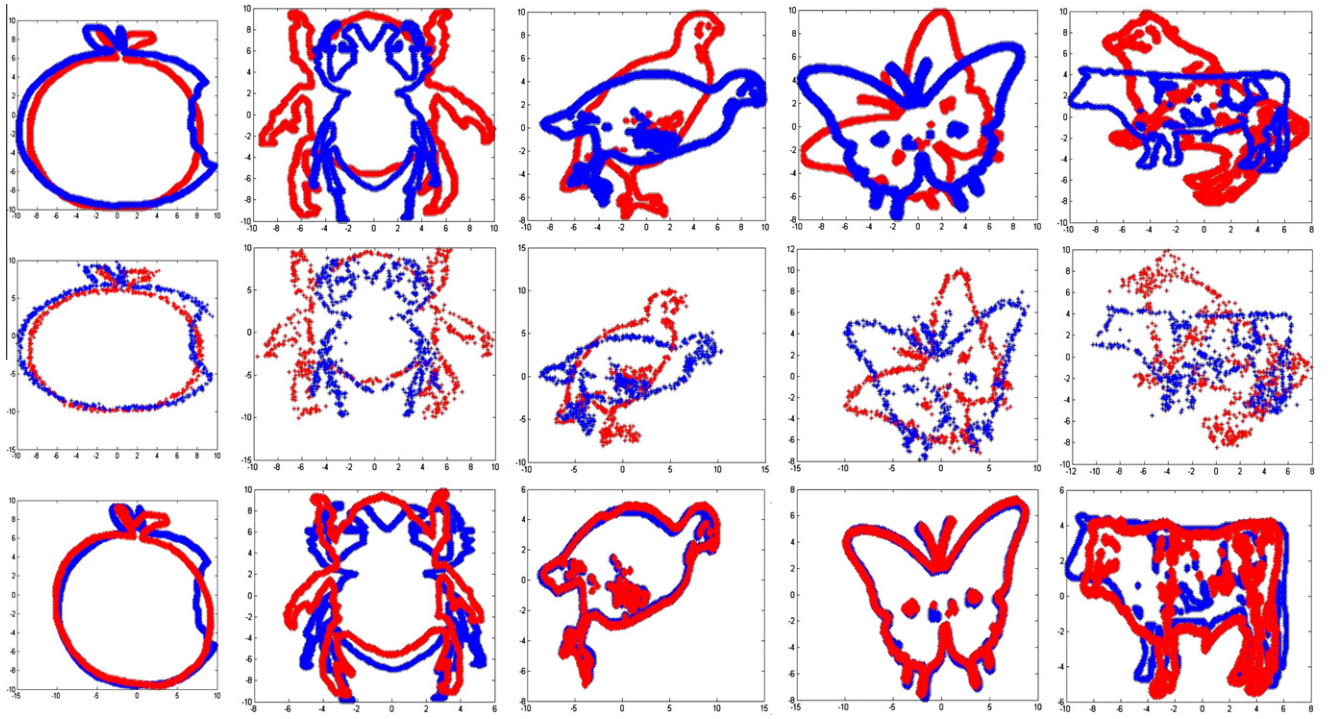


Fig. 4. Shape registration. First row: five pairs of input point sets with unequal number of points. The shapes are centered: the translational component \mathbf{t} is known but not the linear component \mathbf{A} . Second row: the sampled point sets with 700 points. Third row: registration results.

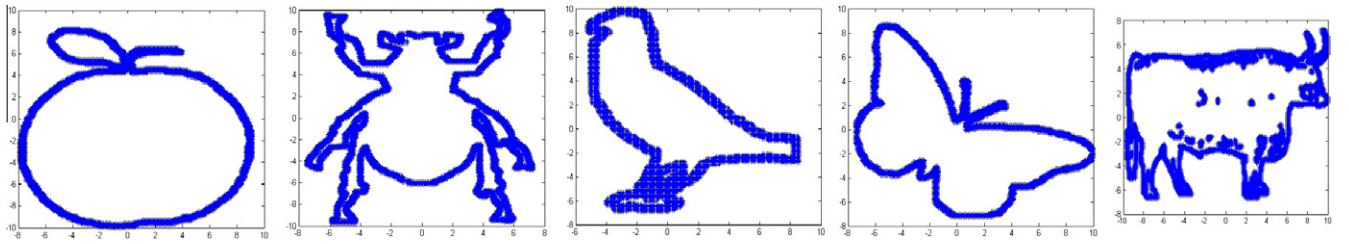


Fig. 5. The five shapes used in the experiments reported in Table 4: Apple, Beetle, Bird, Butterfly, and Cattle.

Table 4

Experimental results with unequal number of points. For each shape and deletion setting ($d\%$ of points deleted), 20 trials with 20 randomly generated affine transformations are performed. The mean relative errors and their variances (in parenthesis) are shown.

%Deletion →	1%	2%	5%	10%	15%
Shape ↓					
Apple	0.02 (0.01)	0.02 (0.02)	0.06 (0.04)	0.11 (0.06)	0.14 (0.13)
Beetle	0.03 (0.03)	0.04 (0.03)	0.8 (0.07)	0.13 (0.12)	0.20 (0.13)
Bird	0.02 (0.02)	0.04 (0.02)	0.06 (0.08)	0.13 (0.10)	0.19 (0.13)
Butterfly	0.03 (0.04)	0.04 (0.05)	0.08 (0.06)	0.13 (0.11)	0.15 (0.13)
Cattle	0.03 (0.03)	0.03 (0.05)	0.05 (0.07)	0.11 (0.11)	0.16 (0.12)

the estimated (linear component) affine transformation and the ground truth. In this experiment, we experimented with five different deletion settings: $d\%$ of points from \mathcal{P} are randomly chosen and deleted for $d = 1, 2, 5, 10$ and 15. For each d , we run the registration algorithm 20 times with 20 randomly generated affine transformations. The mean and variance estimated from the 20 relative errors for all the five shapes used in the experiment are tabulated in Table 4. The results show that the random point deletion method does provide reasonably accurate registration results for two point sets with sizes differed by less than 15%. The method is efficient and does not incur noticeable extra running time.

5. Summary and conclusions

We have proposed a novel affine registration algorithm for matching 2D point sets related by an unknown affine transformation. The most prominent feature of the algorithm is that it is entirely algebraic and does not require any optimization. For data points without noise, the algorithm has been shown to be able to recover the exact affine transformation and correspondence. The algebraic and geometric motivations behind the proposed algorithm is both clear and transparent. While the algorithm was originally formulated for point sets containing equal number of points, we have investigated two methods for handling arbitrary point sets: resampling of points from Gaussian mixture models and random deletion of points. The algorithm is efficient and takes only seconds to finish for point sets with hundreds of points. Experimental results on both synthetic and real-world point-set data show that the proposed algorithm is capable of producing good registration results using noisy inputs.

Acknowledgments

We thank the reviewers for their comments and suggestions. In particular, the idea of using random point deletion to deal with unequal number of points was suggested by one of the reviewers.

References

- [1] G. Scott, C. Lonquet-Higgins, An algorithm for associating the features of two images, *Proceedings of Royal Society of London B244* (1991) 21–26.
- [2] R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003.
- [3] T.F. Cootes, G.J. Edwards, C.J. Taylor, Active appearance models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (6) (2001) 681–685.
- [4] M. Jenkinson, S. Smith, A global optimisation method for robust affine registration of brain images, *Medical Image Analysis* 5 (2) (2001) 143–156.
- [5] F. Bookstein, Principal warps: thin-plate splines and the decomposition of deformations, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11 (6) (1989) 567–585.
- [6] R. Szeliski, S. Lavalley, Matching 3D anatomical surfaces with non-rigid deformations using octree splines, *International Journal of Computer Vision* 18 (2) (1996) 171–186.
- [7] D. Rueckert, L. Sonoda, C. Hayes, D. Hill, M. Leach, D. Hawkes, Non-rigid registration using free-form deformations: application to breast MR images, *IEEE Transactions on Medical Imaging* 18 (8) (1999) 712–721.
- [8] H. Chui, A. Rangarajan, A new algorithm for non-rigid point matching, in: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 44–51.
- [9] B. Horn, Closed-form solution of absolute orientation using unit quaternions, *Journal of the Optical Society of America A: Optics, Image Science and Vision* 4 (4) (1987) 629–642.
- [10] P.J. Besl, N.D. McKay, A method for registration of 3-d shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (2) (1992) 239–266.
- [11] A. Fitzgibbon, Robust registration of 2d and 3d point sets, *Image and Vision Computing* 21 (13–14) (2003) 1145–1153.
- [12] S. Gold, C.-P. Lu, S.P.A. Rangarajan, E. Mjolsness, New algorithms for 2d and 3d point matching: pose estimation and correspondence, in: *Advances in Neural Information Systems*, 1994, pp. 957–964.
- [13] H. Lawson, M. Michelsohn, *Spin Geometry*, Princeton University Press, 1990.
- [14] H. Baird, *Model-Based Image Matching Using Location*, MIT Press, 1984.
- [15] P. Besl, R. Jain, Three dimensional object recognition, *ACM Computing Surveys* 17 (1) (1985) 75–145.
- [16] E. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints*, MIT Press, 1990.
- [17] G. Hinton, C. Williams, M. Revow, Adaptive elastic models for hand-printed character recognition, in: *Advances in Neural Information Systems*, 1992, pp. 512–519.
- [18] D. Huttenlocher, G. Klanderman, W. Rucklidge, Comparing images using the Hausdorff distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15 (9) (1993) 850–863.
- [19] J. Feldmar, N. Ayache, Rigid, affine and locally affine registration of free-form surfaces, *International Journal of Computer Vision* 18 (2) (1996) 99–119.
- [20] D. Metaxas, E. Koh, N. Badler, Multi-level shape representation using global deformation and locally adaptive finite elements, *International Journal of Computer Vision* 25 (1) (1997) 49–61.
- [21] T. Hofmann, J. Buhmann, Pairwise data clustering by deterministic annealing, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (1) (1997) 1–14.
- [22] A. Cross, E. Hancock, Graph matching with a dual-step EM algorithm, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (11) (1998) 1236–1253.
- [23] Z. Yang, F. Cohen, Cross-weighted moments and affine invariants for image registration and matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (8) (1999) 804–814.
- [24] J. Flusser, T. Suk, Pattern recognition by affine moment invariants, *Pattern Recognition* 26 (1) (1993) 167–174.
- [25] H. Suesse, W. Ortmann, Robust matching of affinely transformed objects, in: *Proceedings of International Conference on Image Processing*, vol. 2, 2003, pp. 375–378.
- [26] K. Voss, H. Suesse, Affine point pattern matching, in: *Proceedings of DGAM Symposium*, vol. 2191, 2001, pp. 155–162.
- [27] Z. Zhang, Iterative point matching for registration of free-form curves and surfaces, *International Journal of Computer Vision* 13 (2) (1994) 119–152.
- [28] H. Chui, A. Rangarajan, A new point matching algorithm for non-rigid registration, *Computer Vision and Image Understanding* 89 (2–3) (2003) 114–141.
- [29] L. Shapiro, J. Brady, Feature-based correspondence: an eigenvector approach, *Image and Vision Computing* 10 (5) (1992) 283–288.
- [30] T. Cootes, C. Taylor, D. Cooper, J. Graham, Active shape models: their training and application, *Computer Vision and Image Understanding* 61 (1) (1995) 38–59.
- [31] M. Carcassoni, E. Hancock, Spectral correspondence for point pattern matching, *Pattern Recognition* 36 (1) (2003) 193–204.
- [32] M.-K. Hu, Visual pattern recognition by moment invariants, *IEEE Transactions on Information Theory* 8 (2) (1962) 179–187.
- [33] T. Reiss, The revised fundamental theorem of moment invariants, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (8) (1991) 830–834.
- [34] M. Hagedoorn, R. Velkamp, Reliable and efficient pattern matching using an affine invariant metric, *International Journal of Computer Vision* 21 (2–3) (1999) 203–225.
- [35] D. Huttenlocher, Fast affine point matching: an output-sensitive method, in: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1991, pp. 263–268.
- [36] A. Goshtasby, G. Stockman, Point pattern matching using convex hull edges, *IEEE Transactions on System, Man and Cybernetics* 15 (1985) 631–637.
- [37] Z. Yang, F. Cohen, Image registration and object recognition using affine invariants and convex hulls, *IEEE Transactions on Image Processing* 8 (7) (1999) 934–946.
- [38] C. Gope, N. Kehtarnavaz, Affine invariant comparison of point-sets using convex hulls and Hausdorff distances, *Pattern Recognition* 40 (1) (2007) 309–320.
- [39] S. Gold, A. Rangarajan, C. Liu, S. Pappu, E. Mjolsness, New algorithms for 2d and 3d point matching: pose estimation and correspondence, *Pattern Recognition* 31 (8) (1998) 1019–1031.
- [40] S. Sclaroff, A. Pentland, Modal matching for correspondence and recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17 (6) (1995) 545–561.
- [41] S. Chang, F. Cheng, W. Hsu, G. Wu, Fast algorithm for point pattern matching: invariant to translations, rotations and scale changes, *Pattern Recognition* 30 (2) (1997) 311–320.
- [42] S. Umeyama, Least-squares estimation of transformation parameters between two point-patterns, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (4) (1991) 376–380.
- [43] P. Van Wamelen, Z. Li, S. Iyengar, A fast expected time algorithm for the 2-D point pattern matching problem, *Pattern Recognition* 37 (8) (2004) 1699–1711.
- [44] G. Hjaltason, M. Ray, H. Samet, I. Weiss, Using spatial sorting and ranking in model-based object recognition, in: *Proceedings of International Conference on Pattern Recognition*, 1998, pp. 1347–1349.
- [45] F. Murtagh, A new approach to point pattern matching, *Publications of the Astronomical Society of the Pacific* 104 (674) (1992) 301–307.
- [46] B. Luo, E. Hancock, A unified framework for alignment and correspondence, *Computer Vision and Image Understanding* 92 (2003) 26–55.
- [47] D. Huttenlocher, S. Ullman, Recognizing solid objects by alignment with an image, *International Journal of Computer Vision* 5 (2) (1990) 195–212.
- [48] H. Wolfson, I. Rigoutsos, Geometric hashing: an overview, *IEEE Computational Science and Engineering* 4 (4) (1997) 10–21.
- [49] B. O'Neill, *Elementary Differential Geometry*, Academic Press, 1997.
- [50] B. Jian, B. Vemuri, A robust algorithm for point set registration using mixture of Gaussians, in: *Proceedings of IEEE International Conference on Computer Vision*, 2005, pp. 1246–1251.
- [51] R. Duda, P. Hart, *Pattern Classification*, Wiley-Interscience, 2000.