An Algebraic Approach to Affine Registration of Point Sets

Jeffrey Ho
Department of CISE
University of Florida
Gainesville, FL, USA

Adrian Peter
Northrop Grumman
Melbourne, FL, USA

Anand Rangarajan
Department of CISE
University of Florida
Gainesville, FL, USA

Ming-Hsuan Yang
EECS
University of California
Merced, CA, USA

Abstract

This paper proposes a new affine registration algorithm for matching two point sets in $\mathbb{R}^2$ or $\mathbb{R}^3$. The input point sets are represented as probability density functions, using either Gaussian mixture models or discrete density models, and the problem of registering the point sets is treated as aligning the two distributions. Since polynomials transform as symmetric tensors under an affine transformation, the distributions' moments, which are the expected values of polynomials, also transform accordingly. Therefore, instead of solving the harder problem of aligning the two distributions directly, we solve the softer problem of matching the distributions' moments. By formulating a least-squares problem for matching moments of the two distributions up to degree three, the resulting cost function is a polynomial that can be efficiently optimized using techniques originated from algebraic geometry: the global minimum of this polynomial can be determined by solving a system of polynomial equations. The algorithm is robust in the presence of noises and outliers, and we validate the proposed algorithm on a variety of point sets with varying degrees of deformation and noise.

1. Introduction

This paper proposes a new affine registration algorithm for two point sets in $\mathbb{R}^2$ and $\mathbb{R}^3$ based on solving a system of polynomial equations. Let $\mathbf{S} = \{s_1, \cdots, s_m\}, \mathbf{T} = \{t_1, \cdots, t_n\}$ denote the two point sets for which we seek an affine transform $\mathbf{A}$ that best matches them. Similar to several recent point registration algorithms (e.g., [20, 11]), the point sets $\mathbf{S}, \mathbf{T}$ are assumed to be independently sampled data points according to some probability density functions (PDFs) $\mathcal{P}_\mathbf{S}, \mathcal{P}_\mathbf{T}$, respectively. The main idea of this paper is to solve the registration problem by matching the corresponding moments of the two distributions. Since the moments are the expected values of polynomials, the matching cost function will also be a (multivariate) polynomial, for which there exist now efficient and powerful optimization techniques for computing the *global* minimum (e.g., [5, 14]).

More specifically, any affine transform $\mathbf{A}$ in $\mathbb{R}^k$ induces a linear transform $\mathbf{A}^*$ on the (linear) space of multivariate polynomials on $\mathbb{R}^k$ by the familiar rule: if p is a polynomial, its transformation $\mathbf{A}^*(p)$ under $\mathbf{A}^*$ is the polynomial whose value at a point $x \in \mathbb{R}^k$ is the value of p at the point $\mathbf{A}(x)$,

$$\mathbf{A}^*(p)(x) = p(\mathbf{A}(x)).$$

If the two density functions are related exactly by an affine transform $\mathbf{A}$[1], we have the relation

$$|\det(\mathbf{A})| \mathcal{P}_\mathbf{T}(\mathbf{A}(x)) = \mathcal{P}_\mathbf{S}(x), \tag{1}$$

where $|\det(\mathbf{A})|$ is the absolute value of the determinant of $\mathbf{A}$, which is also the determinant of the Jacobian matrix associated to the transform $\mathbf{A}$. Note that the presence of the determinant is required to ensure that the integral of the density function $\mathcal{P}_\mathbf{T}$ over $\mathbb{R}^k$ is indeed one. For example, a zero mean and unit variance normal distribution is transformed under an linear transform $\mathbf{A}$ into a zero mean normal distribution with variance $\mathbf{A}\mathbf{A}^\top$, and the determinant $|\det(\mathbf{A})|$ in Equation 1 comes from the normalization constant of the transformed density function [1].

It then follows directly from the equation above that the expected values of a polynomial f and its transformation $\mathbf{A}^*(f)$ with respect to their corresponding distributions are the same:

$$\int_{\mathbb{R}^k} \mathbf{A}^*(f)\, \mathcal{P}_\mathbf{S}\, dx = \int_{\mathbb{R}^k} f(\mathbf{A}(x))\, \mathcal{P}_\mathbf{T}(\mathbf{A}(x)) |\det(\mathbf{A})|\; dx$$

where dx is the volume element in $\mathbb{R}^k$, and a simple change of variables shows that

$$\int_{\mathbb{R}^k} \mathbf{A}^*(f)\, \mathcal{P}_\mathbf{S}\, dx = \int_{\mathbb{R}^k} f(y)\, \mathcal{P}_\mathbf{T}(y)\, dy.$$

[1] For example, in $\mathbb{R}^2$, this means that the two random variables x, y are subject to the affine transform: $x \rightarrow ax + by + e, y \rightarrow cx + dy + f$.

If $\mathbf{A}^*(f)$ is a linear combination of some basis polynomials $g_1, \cdots, g_l$,

$$\mathbf{A}^*(f) = a_1\, g_1 + \cdots + a_l g_l,$$

we have

$$\int_{\mathbb{R}^k} f\, \mathcal{P}_{\mathbf{T}} dx = \sum_{i=1}^{l} a_i \int_{\mathbb{R}^k} g_i\, \mathcal{P}_{\mathbf{S}}\, dx. \tag{2}$$

If the basis polynomials $g_i, 1 \le i \le l$ are the monomials, their integrals above are the **moments** [1] of the distributions $\mathcal{P}_{\mathbf{S}}$. Furthermore, if f and g_i are appropriately chosen, the coefficients a_i, $1 \le i \le l$ will be polynomials in the entries of the matrix $\mathbf{M}_{\mathbf{A}}$ representing the affine transform $\mathbf{A}$, and the equation above provides one polynomial constraint for $\mathbf{M}_{\mathbf{A}}$. With a collection of h linearly independent polynomials, $\{f_1, \cdots, f_h\}$, we have h polynomial constraints, which can be used to formulate the matching cost function:

$$\mathcal{E}(\mathbf{A}) = \sum_{i=1}^{h} \omega_i\, \mathbf{MM}_{f_i}^2, \tag{3}$$

where

$$\mathbf{MM}_{f_i}(\mathbf{A}) = \int_{\mathbb{R}^k} f_i\, \mathcal{P}_{\mathbf{T}} d\omega - \sum_{j=1}^{k} a_j^i \int_{\mathbb{R}^k} g_j\, \mathcal{P}_{\mathbf{S}}\, d\omega,$$

and $\omega_i \ge 0$ are the weights. That is, instead of matching points directly, we match the moments, and the affine transform $\mathbf{A}$ is computed as the global minimum of the cost function $\mathcal{E}$, which is a polynomial.

We do not claim that the idea of point registration by matching moments is new. In fact, this idea was already implicit in the early work of [16, 17, 2]. In these seminal work on rigid registration published in the early 90s, the rigid registration is accomplished by matching principal directions computed from the point sets. This can be easily shown to be equivalent to matching the quadratic moments of the discrete density functions associated to the point sets,

$$\mathcal{P}_{\mathbf{S}} = \frac{1}{m}\sum_{i=1}^{m} \delta_{s_i}, \;\; \mathcal{P}_{\mathbf{T}} = \frac{1}{n}\sum_{i=1}^{n} \delta_{t_i}.$$

Algebraically, matching quadratic moments was tractable back in the early 90s because completing squares is relatively easy and global minimization is straightforward. For moments with degrees greater than two, the coefficients a_j^i and hence the cost function $\mathcal{E}$ starts to become cumbersome to compute by hand and its optimization requires more elaborate techniques. Two developments in the interim period have made it possible now to explore the possibility of matching higher-degree moments for point registrations. First, software packages such as MAPLE that do symbolic computations have matured considerably and become widely available. This makes computing the cost function $\mathcal{E}$ almost effortless. Second and more importantly, there are now efficient and robust techniques available for computing the global minimums of polynomials [5, 14]. This is very important because it is difficult to guarantee the quality of the solution if it is only a local minimum of the cost function.

Needless to say, affine registration has been studied quite extensively in vision literature. Most of the published algorithms require some kind of optimization either directly on $\mathbb{R}^k$ or on some relevant Lie group such as $\mathbf{SO}(k)$. For this part, gradient descent has always been the method of choice. While many of these algorithms perform well most of the time, it is virtually impossible to guarantee convergence of these algorithms to the global minimum of their cost functions. However, by formulating a polynomial cost function, we can indeed guarantee that our solution will be the global minimum of the cost function. Furthermore, our approach makes a good intuitive sense because distributions can be characterized by their moments, and a good affine transform between the two distributions should match the moments reasonably well. In particular, compared to moments of higher-degrees, the first few moments (low-degree moments) are much more important because they encode certain global geometric properties of the distributions that are important for perceptions. For example, the linear moments are related to the centroid, the quadratic moments are related to the variance and the cubic moments are related to skewness [1]. Therefore, even if the affine transform associated to the global minimum of our cost function is not the true affine transform, we can still expect that the true affine transform should be near our solution because it will most likely match the moments almost as well as our solution.

2. Affine Registration Algorithm

In this section, we detail the proposed affine registration algorithm. We will first describe how the polynomials transformed under a general affine transformation. This is then followed by the discussions on how to cast the registration problem into the form that can be solved using polynomial optimization.

Let $\mathbf{S}$ and $\mathbf{T}$ as above denote two point sets in $\mathbb{R}^k$, and our problem is to estimate an affine transformation $\mathbf{A}$ that best approximates $\mathbf{T}$ by the image of $\mathbf{S}$ under $\mathbf{A}$[2]. In the following discussions, we will consider only point sets in $\mathbb{R}^2$ since the $\mathbb{R}^3$ case is almost exactly the same except the algebra is messier because of more variables. For the moment, we will ignore the translational component of $\mathbf{A}$ and focus on its linear component

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \tag{4}$$

[2] $k = 2, 3$ are the cases of interest.

2.1. Induced Linear Transformations

As a linear transform of $\mathbb{R}^2$, $\mathbf{A}$ induces a family of linear transformations on the polynomials. Specifically, for each integer $d \geq 1$, $\mathbf{A}$ induces a linear transformation $\mathbf{A}_d^*$ on the space of polynomials of degree d in x and y. This can be seen immediately using abstract linear algebra because homogeneous polynomials of degree d are symmetric tensors in the d-fold tensor product space[3] $V^{\otimes d} \equiv V \otimes V \otimes \cdots \otimes V$, where V is the vector space $\mathbb{R}^2$. Any linear transformation of the base vector space V will induce a linear transformation for each $V^{\otimes d}$ that maps symmetric tensors to symmetric tensors, i.e., linear transformations of homogeneous polynomials. The important point is that once a matrix representation for $\mathbf{A}$ is given, all the linear transformations $\mathbf{A}_d^*$ can be written down in matrix form such that the entries of the matrix for $\mathbf{A}_d^*$ are degree d polynomials in the entries of the matrix for $\mathbf{A}$.

Instead of citing standard references (e.g., [10]) to illustrate the above, we will work out one example in detail. Consider the affine transformation $\mathbf{A}$ in Equation 4. Under $\mathbf{A}$, the linear monomials x, y are transformed into the following two linear monomials:

$$x \rightarrow ax + by, \qquad y \rightarrow cx + dy.$$

Since these two monomials x, y form a basis for the space of linear polynomials on $\mathbb{R}^2$, the above two transformations define a linear transformation $\mathbf{A}_1^*$ on the space of linear polynomials, and the matrix representation for $\mathbf{A}_1^*$ using this basis is the transpose of $\mathbf{A}$ in Equation 4. This result can be extended easily to define transformations for monomials of higher degrees using polynomial multiplications. For instance, when $d = 2$, we have the transformations for the following three monomials:

$$\begin{aligned} x^2 &\rightarrow (ax + by)^2 = a^2x^2 + 2abxy + b^2y^2, \\ xy &\rightarrow (ax + by)(cx + dy) = acx^2 + (ad + bc)xy + bdy^2, \\ y^2 &\rightarrow (cx + dy)^2 = c^2x^2 + 2cdxy + d^2y^2. \end{aligned}$$

Since any quadratic polynomial is a linear combination of the above three basis monomials, these transformations allow us to define a linear transformations $\mathbf{A}_2^*$ for the space of quadratic polynomials as above, and its matrix representation is (using this basis)

$$\mathbf{A}_2^* \equiv \begin{pmatrix} a^2 & ac & c^2 \\ 2ab & ad + bc & 2cd \\ b^2 & bd & d^2 \end{pmatrix}.$$

For degree three polynomials, the transformations for the four basis monomials are

$$\begin{aligned} x^3 &\rightarrow a^3x^3 + 3a^2bx^2y + 3ab^2xy^2 + b^3y^3, \\ x^2y &\rightarrow a^2cx^3 + (a^2d + 2abc)x^2y + (2abd + cb^2)xy^2 + b^2dy^3, \\ xy^2 &\rightarrow ac^2x^3 + (c^2b + 2acd)x^2y + (2bcd + ad^2)xy^2 + bd^2y^3, \\ y^3 &\rightarrow c^3x^3 + 3c^2dx^2y + 3cd^2xy^2 + d^3y^3, \end{aligned}$$

and the matrix for $\mathbf{A}_3^*$ is

$$\mathbf{A}_3^* \equiv \begin{pmatrix} a^3 & a^2c & ac^2 & c^3 \\ 3a^2b & a^2d + 2abc & c^2b + 2acd & 3c^2d \\ 3ab^2 & 2abd + cb^2 & 2bcd + ad^2 & 3cd^2 \\ b^3 & b^2d & bd^2 & d^3 \end{pmatrix}.$$

For any degree $d \geq 1$, the space of homogeneous polynomials of degree d has dimension $d + 1$ as there are $d + 1$ linearly independent degree d monomials $x^d\, x^{d-1}y, \cdots, xy^{d-1}, y^d$. We will denote g_d^i for $1 \leq i \leq d+1$ these $d+1$ basis monomials of degree d, and the above discussions can be summarized succinctly by the following equation

$$\mathbf{A}_d^*(g_d^i) = \sum_{j=1}^{k} A_{dj}^i\, g_d^j, \tag{5}$$

where A_{dj}^i denote the entries of the matrix for $\mathbf{A}_d^*$. Note that each A_{dj}^i is indeed a degree d homogeneous polynomial in the entries of the matrix $\mathbf{A}$ (a, b, c and d).

2.2. Matching Moments

Applying Equations 2 and 5, we have

$$\int_{\mathbb{R}^k} g_d^i\, \mathcal{P}_{\mathbf{T}} dx = \sum_{j=1}^{k} A_{dj}^i \int_{\mathbb{R}^k} g_d^j\, \mathcal{P}_{\mathbf{S}}\, dx. \tag{6}$$

Each integral in the equation above defines one particular moment of degree d for its corresponding density function [1]. Therefore, if we assume that all moments $\int_{\mathbb{R}^k} g_d^i\, \mathcal{P}_{\mathbf{S}}\, dx$, $\int_{\mathbb{R}^k} g_d^i\, \mathcal{P}_{\mathbf{T}}\, dx$ of a given degree exist, the above equation then provides us with one constraint for $\mathbf{A}$. In particular, since we assume that the distributions $\mathcal{P}_{\mathbf{S}}, \mathcal{P}_{\mathbf{T}}$ are either the discrete densities or Gaussian mixtures, all of their moments exist and is finite: for the former, it is just the average of the values of the polynomial evaluated at the points and for the latter, the moments can be directly computed using the moment generating function for a multivariate normal distribution $\mathbf{N}(\mu; \Sigma)$ with mean μ and covariance matrix Σ [1]:

$$\mathbf{M}(t) = \exp(\mu^\top \mathbf{t} + \frac{1}{2}\mathbf{t}^\top \Sigma \mathbf{t}),$$

where $\mathbf{t}$ is the column vector $[x\, y]^\top$. The specific formula we need is that the moments of $\mathbf{N}(\mu; \Sigma)$ can be computed by

$$\int_{\mathbb{R}^k} x^i y^j\, \mathbf{N}(\mu; \Sigma)\, dx = \frac{\partial \mathbf{M}}{\partial^i x \partial^j y}(0),$$

which allows us to quickly compute the moments for a Gaussian mixture by taking the (weighted) sum of the moment of each Gaussian component.

Finally, we denote $\mathbf{I}_d^{\mathbf{S}}, \mathbf{I}_d^{\mathbf{T}}$ the vector of degree d moments of $\mathcal{P}_{\mathbf{S}}, \mathcal{P}_{\mathbf{T}}$, respectively. These vectors are obtained by vertically stacking the $d + 1$ moments formed by integrating

[3] These are the rank-d tensors.

the $d+1$ basis monomials g_d^i, and accordingly, $\mathbf{I}_d^{\mathbf{S}}, \mathbf{I}_d^{\mathbf{T}}$ are vectors of length $d+1$. Equation 6 can now be put into a very succinct form

$$\mathbf{I}_d^{\mathbf{T}} = \mathbf{A}_d^* \mathbf{I}_d^{\mathbf{S}} \tag{7}$$

for all $d > 0$. Therefore, for a point registration problem, once the PDFs $\mathcal{P}_{\mathbf{S}}, \mathcal{P}_{\mathbf{T}}$ have been computed, we can try to minimize the following cost function

$$\mathcal{E}(\mathbf{A}) = \sum_{d=1}^{D} \omega_d \, \|\mathbf{I}_d^{\mathbf{T}} \; - \; \mathbf{A}_d^* \, \mathbf{I}_d^{\mathbf{S}}\|^2, \tag{8}$$

where $\omega_d > 0$ are the weights, and D gives the upper bound on the degrees of the moments used in the matching. Note that all the vectors $\mathbf{I}_d^{\mathbf{S}}, \mathbf{I}_d^{\mathbf{T}}$ are known and their dimensions depend on the degree d. For a fixed D, Equation 8 is a polynomial of degree $2D$ in the entries of $\mathbf{A}$.

We remark that in general we can assume that all the linear moments vanish, and this corresponds to centering the point set with respect to it own centroid. In $\mathbb{R}^k$, there are $C(k+1, 2) = k(k+1)/2$ quadratic basis moments. This number is smaller than the dimension of $\mathbf{GL}(k)$, which is k^2. In particular, if we match only up to quadratic moments in Equation 8, the resulting optimization problem will not have finite number of solutions but a continuous family of solutions. In terms of linear algebra, this corresponds to the fact that given two covariance (symmetric and positive-definite) matrices, we can always match the two matrices using affine transformations in $\mathbf{GL}(k)$. Therefore, this shows that it is not enough to match only quadratic moments. In the experiments below, we will match the moments up to degree three, and a simple dimension counting will show that for a general pair of point sets, the solution (global minimum of $\mathcal{E}$) will be finite in number.

2.3. Polynomial Optimization

Given a multivariate polynomial $P(x_1, \cdots, x_n)$, a direct method for computing its global minimum is to enumerate all possible critical points of P, and the global minimum would be among these critical points. The critical points of P are the points at which the gradient of P vanishes

$$\nabla P \; = \; (\frac{\partial P}{\partial x_1}, \cdots, \frac{\partial P}{\partial x_n}) = 0.$$

This provides us with a system of n polynomial equations with degree one less than the degree of P. This system of equations can be solved by computing a Gröbner basis for the polynomial ideal generated by the n polynomials $\frac{\partial P}{\partial x_1}, \cdots, \frac{\partial P}{\partial x_n}$ [5]. In order to compute the Gröbner basis efficiently, we need to define a monomial order carefully (details omitted), and there exist several software packages (e.g. MAPLE) that can efficiently compute the Gröbner basis given polynomial inputs and the monomial order. The same method for solving polynomial equations has been applied previously to solve other vision problems such as triangulation and camera calibration (e.g. [19, 13]). However, we remark that our problem is significantly easier than theirs because there is no "denominators" in our cost function and the application of the Gröbner basis here is much more straightforward.

3. Related Work

Using probability distributions in point set registration has become quite popular in recent years, e.g., [4, 20, 11, 8]. Replacing the point sets with their associated distributions provides a more robust and principled way of dealing with outliers and noises. While these cited papers deal with the more general non-rigid registration, the affine registration is nevertheless an important component in these algorithms as it is typically the crucial first step in aligning the point sets. In [20], the matching cost function is the KL-divergence between the two distributions and in [11], it is the L^2-distance between the distributions. Affine registration is handled using gradient descent in these papers, and in [11], the closed-form formula for the gradient is known and this makes the optimization more efficient when compared with the case (e.g., [20]), where the gradients have to be computed numerically.

There are registration algorithms that do not require minimization using gradient descent. As mentioned in the introduction, the methods proposed in [16, 17, 2] are basically equivalent to matching quadratic moments. To the best of our knowledge, there does not seem to have been any significant follow-up of these works that tackles the point set registration problem using higher-degree moments, except perhaps [9]. However, the algorithm proposed in [9] requires complex multiplications and this limits its applicability to point sets in $\mathbb{R}^2$. Furthermore, the invariants matched there are generated by evaluating elementary symmetric functions on the points, which, although similar, cannot be interpreted as (discrete) moments. Iterative closest point (ICP) (e.g., [15, 18, 7]) is another class of registration algorithms that does not require continuous optimization (the optimization can be solved quickly using linear algebra once the correspondences are known). While this algorithm has been adopted widely and in many cases perform brilliantly, it is difficult to prove the algorithm's convergence and at the same time, nothing much can be said about the quality of its solution in general.

Finally, the polynomial optimization algorithms used in this paper have already made their appearances in several important vision papers published in the past few years, e.g., [12, 19, 13]. Polynomial cost functions abound in multi-view geometry problems, ranging from three-view triangulation, 2D homography estimation to camera calibration (with radial distortion). We note however that our context is different from theirs in that the point correspondences are known in these works and their algorithms output the optimal transformations according to the known correspondences. In our case, the correspondences are not known on the point-level (we are trying to compute the correspondences) but on the moment-level. That is, the moment vectors $\mathbf{I}^{\mathbf{T}}_d, \mathbf{I}^{\mathbf{S}}_d$ of the same degree correspond under the induced linear transform $\mathbf{A}^*_d$ (Equation 7).

4. Experiments

For lack of space, we will report only two sets of experimental results. More (including 3D) results are available online [4]. Our goal in these two experiments is to demonstrate that the proposed algorithm is indeed robust against noises and outliers. In the first set of experiments, we randomly generate point sets and affine transformations in $\mathbb{R}^2$. Various different amounts of noise are added to the point sets and we study the behavior of the algorithm under different noise settings. Second, we apply the algorithm to 2D point sets that are extracted from MPEG shape database, and we show several affine registration results between pairs of similar (but not the same) shapes. We have implemented the algorithm using MATLAB and MAPLE without any optimization. The sizes of the point sets in the experiments range from 100 to more than 4000, and both experiments were run on a DELL desktop with a single 3.1GHz processor.

4.1. Affine Registration in $\mathbb{R}^2$

In this set of experiments, our aim is to give a qualitative as well as quantitative analysis on the accuracy and robustness of the proposed method. We report our experimental results on synthetic data using point sets of different sizes with various different noise settings. Tables 1 summarizes the experimental results. The algorithm is tested in four different sizes, 100, 200, 500 and 1000, and five different noise settings, $0\%, 1\%, 2\%, 5\%, 10\%$. For each pair of point set size and noise setting, we ran 100 trials, each with a randomly generated non-singular matrix $\mathbf{A}$ and a point set of the given size. In trials with $x\%$ noise setting, we add a uniform random noise ($\pm x\%$) to each coordinate of every point independently. Let $\mathbf{A}'$ denote the estimated matrix. A point $s \in \mathbf{S}$ is matched to the point $t \in \mathbf{T}$ if $t = \min_{t_i \in \mathbf{T}} \mathbf{dist}(\mathbf{A}'s, t_i)$. For each trial, we report the percentage of mismatched points and the relative error of the estimated matrix $\mathbf{A}'$: $\frac{\|\mathbf{A}'-\mathbf{A}\|_F}{\|\mathbf{A}\|_F}$, using the Frobenius norm.

As the point sets are randomly generated in this experiment, Gaussian mixtures are clearly not suited for modelling these point sets, and we use the discrete densities instead. We use moments up to degree three, and the matching cost function $\mathcal{E}(A)$ is a degree six polynomial with four variables. The Gröbner basis method is used to solve for the affine transformation.

4.2. Matching Shapes in $\mathbb{R}^2$

In the second set of experiments, we work with shapes from the MPEG shape database. In this shape database, there are seventy different shape categories and each category has twenty different shapes. We apply the proposed algorithm to affine register shapes in the same category. In this experiment, the shapes are represented as point sets with sizes ranging from 2000 to about 5000 points. A Gaussian mixture model is estimated for each shape using Expectation Maximization algorithm [6, 3] with the number of mixture components ranging from 15 to 33. The matching cost function $\mathcal{E}$ is a polynomial with degree six (using all the moments up to degree three), and the Gröbner basis method is used to solve for the affine registration. Figure 1 displays several examples of successful affine registrations using the proposed algorithm.

5. Conclusion

We have proposed a new affine registration algorithm for two point sets in $\mathbb{R}^2$ or $\mathbb{R}^3$ based on matching the moments of two probability density functions. The matching cost function turns out to be a polynomial, and its global minimum can be efficiently determined by solving a system of polynomial equations. Experimental results have demonstrated that this new approach is indeed viable, and the algorithm is capable of producing good solutions even in the presence of significant amount of noises and outliers.

References

[1] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. John Willey & Sons, 1984. 1, 2, 3

[2] P. J. Besel and H. D. Mckay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 2, 4

[3] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., 1995. 5

[4] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, volume 2, pages 44–51, 2000. 4

[5] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Springer, 2005. 1, 2, 4

[6] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statistical Society. Series B*, 39(1):1–38, 1977. 5

[7] A. W. Fitzgibbon. Robust registration of 2d and 3d point sets. *Computer Vision and Image Understanding*, 2(13):1145–1153, 2003. 4

[8] J. Goldberger, S. Gordon, and H. Greenspan. An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *Proc. Int. Conf. on Computer Vision*, pages 487–493, 2003. 4

[4] http://www.cise.ufl.edu/~jho/AffineRegistration/

Table 1. **Experimental Results** For each pair of point set size and noise setting, one hundred trials, each with a randomly generated matrix $\mathbf{A}$ and a point set, are performed. The averaged relative error of the estimated matrix and the percentage of mismatched points (in parenthesis) are shown.

Point Set Size →	100	200	500	1000
Noise ↓	Matrix Error	Matching Error	Matrix Error	Matching Error
0%	0.0 (0)	0.0 (0)	0.01 (0)	0.01 (0)
1%	0.0 (0)	0.0 (0)	0.02 (0)	0.02 (0)
2%	0.0 (0)	0.0 (0)	0.003 (0.1)	0.02 (0.1)
5%	0.01 (1)	0.01 (1)	0.01 (1)	0.03 (1)
10%	0.037 (2)	0.038 (2)	0.041 (2)	0.042 (3)

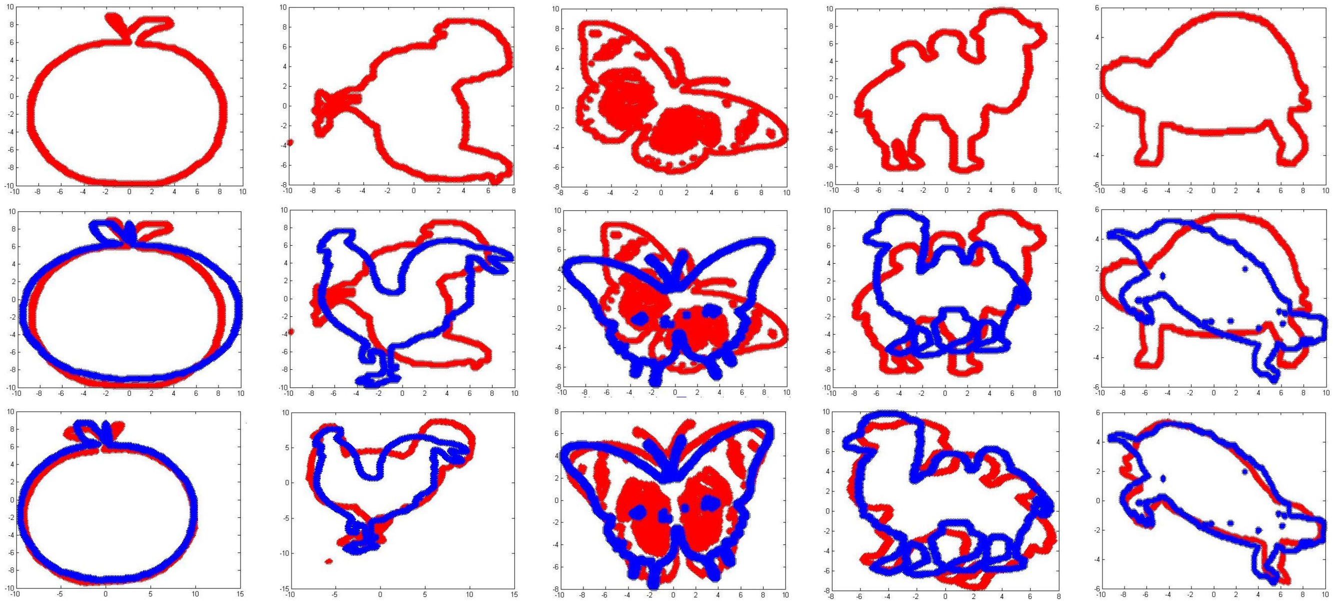

Figure 1. **Matching 2D Shapes** Results of matching shapes from five different shape categories in the MPEG Shape database. The source shapes are displayed in red in the first row. The source and target shapes (in blue) are displayed together in the second row. The matching results are shown in the last row.

[9] J. Ho, M.-H. Yang, A. Ranganrajan, and B. Vemuri. A new affine registration algorithm for 2d point sets. In *WACV*, 2007. 4

[10] M. Itskov. *Tensor Algebra and Tensor Analysis for Engineers*. Springer, 2009. 3

[11] B. Jian and B. Vemuri. A robust algorithm for point set registration using mixture of gaussians. In *Proc. Int. Conf. on Computer Vision*, pages 1246–1251, 2005. 1, 4

[12] F. Kahl and D. Henrion. Globally optimal estimates for geometric reconstruction problems. In *Proc. Int. Conf. on Computer Vision*, pages 978–985, 2005. 4

[13] Z. Kukelova and T. Pajdla. A minimal solution to the autocalibration of radial distortion. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, 2007. 4

[14] J. Lasserre. Global optimization with polynomials and the problem of moment. *SIAM Journ. Optimization*, 11:796–817, 2001. 1, 2

[15] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proc. Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, pages 145–152, 2001. 4

[16] G. Scott and C. Lonquiet-Higgins. An algorithm for associating the features of two images. *Proc. of Royal Society of London*, B244:21–26, 1991. 2, 4

[17] L. Shapiro and J. Brady. Feature-based correspondence: an eigenvector approach. *Image and Vision Computing*, pages 283–288, 1992. 2, 4

[18] G. C. Sharp, S. W. Lee, and D. K. Wehe. ICP registration using invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):90–102, 2002. 4

[19] H. Stewnius, F. Schaffalitzky, and D. Nister. How hard is three-view triangulation really? In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, volume 1, pages 686–693, 2005. 4

[20] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *Proc. European Conf. on Computer Vision*, pages 558–569, 2004. 1, 4