On Planar Point Matching Under Affine Transformation

John E. Hopcroft
Daniel P. Huttenlocher

TR 89-986
April 1989

TECHNICAL REPORT

Department of Computer Science
Cornell University
Ithaca, New York

On Planar Point Matching Under Affine Transformation

John E. Hopcroft
Daniel P. Huttenlocher

TR 89-986
April 1989

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

Research on this paper was supported by DARPA grant N0014-88-K-0591, NSF grant DCM-86-17355, and ONR grant N00014-86-K-0281.

On Planar Point Matching Under Affine Transformation

John E. Hopcroft
Daniel P. Huttenlocher

Computer Science Department
Cornell University

Abstract

An important geometric matching problem in machine vision and robotics is to determine whether there exists an affine transformation (a general linear transformation and a translation) that maps each point of a set A onto a corresponding point of a set B. In the case of matched cardinality point sets, we have developed an optimal $\Theta(n \log n)$ algorithm for determining the existence of such a transformation. The method makes use of the fact that an affine transformation preserves the center gravity of a point set, as well as the ratios of triangle areas.

If $|A| < |B|$ then there can be $O(n^3)$ affine transformations from A to B. In general the number of transformations will be much smaller, so we have developed an *output sensitive* algorithm that runs in time $O(n^2 \log n + tm \log n)$, where $m = |A|$, $n = |B|$, and t depends on the number of transformations. The method relies on the affine properties that intersection points and length ratios along a line are preserved.

1 Introduction

We consider the problem of matching a set of points, A, in the plane to another set, B, under an affine transformation, where $m \leq n$, for $m = |A|$ and $n = |B|$. That is, given two planar point sets A and B the problem is to exhibit any two-dimensional affine transformation, $T : A \rightarrow B$ that maps each point of A onto a point of B. When $m < n$ there are $O(n^3)$ possible matches, because each triple of noncolinear points defines an affine transformation of the plane. In general, however, the number of matches will be small, and thus we seek an *output sensitive* algorithm, that runs in time proportional to the number of transformations from A to B.

The problem of matching planar point sets under an affine transformation has been considered by a number of researchers in computer vision (e.g., [HU] [LSW] [TM]), because the image of a planar surface under projection is reasonably approximated by an affine transformation [Ho]. Related problems of matching under similarity and congruence have been considered in both the computer vision and computational geometry literature (e.g., [AMWW] [Ba]). In this paper we develop new algorithms for the affine matching problem that improve upon the asymptotic time complexity of existing methods. From a practical point of view these algorithms are also straightforward to implement.

When two point sets, A and B, are of equal cardinality an algorithm has been developed by [HT] for determining in average-case linear time whether or not there exists an affine transformation, $T : A \rightarrow B$. The worst-case time bound for the method, however, is quadratic. When the point sets are of unmatched cardinality, a paper by [HU] presents a naive algorithm that matches a given triple of points in A against every triple of points in B. Each such match defines an affine transformation that must be checked to see if it maps all the points of A to corresponding points of B. This can be done by applying the transformation to each point of A and then using a range query to search for a matching point of B. Each range query can be done in $O(\log n)$ time (e.g., as in [Ed]), so the overall running time of the method is $O(n^3 m \log n)$.

In this paper we present an algorithm for the equal cardinality affine matching problem that runs in optimal time $\Theta(n \log n)$, and an algorithm for the case where $m < n$ that runs in time $O(n^2 \log n + tm \log n)$, where t is the number of matches of a given four points in A to any four points in B. In general t is small, but in the worst case it is $\Theta(n^3)$. For example when the points of A form a square and the points of B are on a regularly spaced grid, there are $\Theta(n^3)$ parallelograms in B each of which matches the points of A.

2 Properties of the Affine Transformation

This section briefly describes several properties of affine transformations that are used in the algorithms developed below. These properties hold if and only if a transformation is affine. Derivations of the properties can be found in a number of standard texts (e.g., [Ga] [Kl]) and thus are not presented here.

An affine transformation of the plane, $A : \mathcal{R}^2 \rightarrow \mathcal{R}^2$ can be represented as a nonsingular 2×2 matrix, $\mathbf{L}$, and a translation vector $\mathbf{b} \in \mathcal{R}^2$, such that

$$\mathbf{x}' = \mathbf{L}\mathbf{x} + \mathbf{b},$$

for any $\mathbf{x} \in \mathcal{R}^2$.

Property 1 *An affine transformation of the plane is defined uniquely by three pairs of points. If* $\mathbf{a}$, $\mathbf{b}$ *and* $\mathbf{c}$ *are noncolinear points, and* $\mathbf{a}'$, $\mathbf{b}'$ *and* $\mathbf{c}'$ *are corresponding points, then there exists a unique affine transformation* $A : \mathcal{R}^2 \rightarrow \mathcal{R}^2$, *mapping each of the three given points to its corresponding point.*

Property 2 *Ratios of distances along a line are preserved. If* $\mathbf{a}$, $\mathbf{b}$ *and* $\mathbf{c}$ *are colinear points, and* A *is an affine transformation, then*

$$\frac{|\mathbf{a} - \mathbf{b}|}{|\mathbf{b} - \mathbf{c}|} = \frac{|A(\mathbf{a}) - A(\mathbf{b})|}{|A(\mathbf{b}) - A(\mathbf{c})|},$$

where $|x|$ *is the length of segment* x.

Property 3 *Intersection points are preserved. If* $\mathbf{c}$ *is the point of intersection of two lines* a *and* b, *and* A *is an affine transformation, then* $A(\mathbf{c})$ *is the intersection of the lines* $A(a)$ *and* $A(b)$.

Property 4 *The center of gravity of a set of points is preserved. If* $\mathbf{c}$ *is the center of gravity of a point set,* X, *and* A *is an affine transformation, then* $A(\mathbf{c})$ *is the center of gravity of the point set* $X' = \{\mathbf{x}' | \mathbf{x}' = A(\mathbf{x}), \mathbf{x} \in X\}$.

Property 5 *Ratios cf triangle areas are preserved. If* $\mathbf{a}$, $\mathbf{b}$ *and* $\mathbf{c}$ *are noncolinear points,* $\mathbf{d}$, $\mathbf{e}$ *and* $\mathbf{f}$ *are noncolinear points (not necessarily distinct from* $\mathbf{a}$, $\mathbf{b}$, *and* $\mathbf{c}$*), and* A *is an affine transformation, then*

$$\frac{|\triangle\mathbf{abc}|}{|\triangle\mathbf{def}|} = \frac{|\triangle A(\mathbf{a})A(\mathbf{b})A(\mathbf{c})|}{|\triangle A(\mathbf{d})A(\mathbf{e})A(\mathbf{f})|},$$

where $|x|$ *is the area of triangle* x.

3 Equal Cardinality Point Sets

In order to develop a method for determining whether there is an affine transformation mapping A to B, where $|A| = |B|$, we make use of Property 4, that the center of gravity of a point set is preserved, and Property 5, that the ratios of areas of any two triangles are preserved. The method consists of two stages. The first stage (Algorithm 1) is to compute a canonical form of a point set that is preserved under an affine transformation. The second stage (Algorithm 2) is to compare the canonical forms of two point sets for equality.

To simplify the presentation we initially assume that for each point $\mathbf{x}_i \in X$ the orientation of the segment $\mathbf{cx}_i$ is distinct, where $\mathbf{c}$ is the center of gravity of X.

Algorithm 1 *Given a set X of n points in the plane:*

1. *Compute the center of gravity of X,* $\mathbf{c}$.

2. *Form the list of segments $\mathbf{cx}_i$ for each $\mathbf{x}_i \in X$, determine the orientation of each segment with respect to some fixed orientation (e.g., the x-axis), and sort the segments by orientation yielding a circular list of segments.*

3. *For each pair of successive segments in the circular list, compute the area of the triangle formed by the two segments. That is, the areas $|\triangle \mathbf{cx}_i\mathbf{x}_{i+1}|$, denoted by solid lines in Figure 1.*

4. *For each pair of segments with one intervening segment in the circular list, compute the area of the triangle formed by the two segments. That is, the areas $|\triangle \mathbf{cx}_i\mathbf{x}_{i+2}|$, denoted by dashed lines in Figure 1.*

5. *Compute the pairs of area ratios formed by dividing each area from Step 3 by the corresponding area from Step 4. That is, the ratio pairs*

$$\left(\frac{|\triangle \mathbf{cx}_i\mathbf{x}_{i+1}|}{|\triangle \mathbf{cx}_i\mathbf{x}_{i+2}|}, \frac{|\triangle \mathbf{cx}_{i+1}\mathbf{x}_{i+2}|}{|\triangle \mathbf{cx}_i\mathbf{x}_{i+2}|}\right).$$

   This yields a list of n area ratio pairs.

From Properties 4 and 5 we know that each of the area ratios computed in Algorithm 1 remains unchanged under an affine transformation of a point set, X. Thus the list of area ratio pairs computed in Step 5 is a canonical form for X that is invariant under an affine transformation.

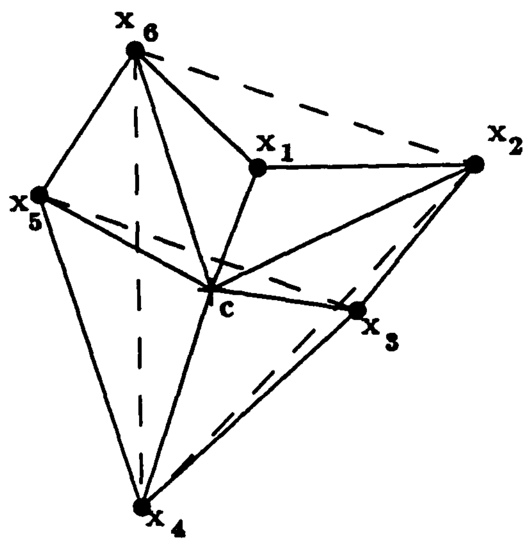

Figure 1: Computing the canonical form: ratios of triangle areas for each pair of successive segments (e.g., $\triangle \mathbf{c x_1 x_2}$) and each pair of segments with one segment intervening (e.g., $\triangle \mathbf{c x_1 x_3}$).

When more than one of the segments, $\mathbf{cx}_i$, share the same orientation, a simple modification to the canonical form suffices. For each duplicated orientation choose some representative point, such as the one farthest from **c**. For any other segment at the same orientation, the length ratio of that segment to the representative segment will be preserved exactly in the case that there is an affine transformation (by Property 2). Thus the canonical form can be computed by Algorithm 1 for the representative points, and then augmented with these length ratios. We now consider how to use the canonical form to determine whether there is an affine transformation from one point set to another.

Algorithm 2 *Given two sets of n points in the plane, A and B:*

1. *Compute the canonical form of A using Algorithm 1.*
2. *Compute the canonical form of B using Algorithm 1.*
3. *Compare the two resulting lists of area ratios for* circular equality *by replicating the list for B twice and using a string matching algorithm to search for the A sequence as a substring of the B sequence.*

To establish that Algorithm 2 correctly decides whether or not there is an affine transformation from A to B we show that the following holds.

Proposition 1 *The canonical forms of two point sets A and B are circularly equal if and only if there exists a unique affine transformation $T : A \rightarrow B$.*

From the construction of the canonical form we know that if there is an affine transformation $T : A \rightarrow B$ then the canonical forms for A and B will be circularly equal (because triangle area ratios are preserved under an affine transformation). To show that the converse is also true we will make use of the following two Lemmas which establish that there is a unique affine transformation from one set of four points to another exactly when two triangle area ratios are preserved.

Lemma 1 *Consider four points in the plane,* $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, *and* $\mathbf{d}$. *If the area ratios of two of the triangles to a third triangle are known, such as*

$$\frac{|\triangle abc|}{|\triangle abd|},$$

and

$$\frac{|\triangle acd|}{|\triangle abd|},$$

then the location of one of the points is defined uniquely with respect to the other three.

This follows straightforwardly from the definition of the area of a triangle. The two known ratios and three known points result in two independent linear equations in two unknowns for the fourth point.

Lemma 2 *Given four points in the plane,* $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, *and* $\mathbf{d}$, *and four corresponding points,* $\mathbf{a}'$, $\mathbf{b}'$, $\mathbf{c}'$, *and* $\mathbf{d}'$, *such that,*

$$\frac{|\triangle abc|}{|\triangle abd|} = \frac{|\triangle a'b'c'|}{|\triangle a'b'd'|} = r_1,$$

and

$$\frac{|\triangle acd|}{|\triangle abd|} = \frac{|\triangle a'c'd'|}{|\triangle a'b'd'|} = r_2,$$

there exists a unique affine transformation mapping $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, *and* $\mathbf{d}$ *to their corresponding points.*

This follows from the previous Lemma, and two of the properties of affine transformations. We know from Property 1 that the correspondence of **a**, **b**, and **c** with **a′**, **b′**, and **c′** defines a unique affine transformation. Given that the two area ratios, r_1 and r_2 are known, by Lemma 1 the position of **d′** is uniquely defined with respect to the other three points. In other words there is a unique transformation mapping **a**, **b**, **c**, and **d** to **a′**, **b′**, **c′**, and **d′**, respectively. From Property 5 this transformation is affine because triangle area ratios are preserved.

It should be noted that Lemmas 1 and 2 only serve to reduce the number of area ratios that must be considered for four points. If all twelve ratios of the four triangle areas defined by four points are preserved by a transformation, then it follows immediately from Property 5 that the transformation must be affine.

Now we return to the problem of showing that if the canonical forms for two point sets, A and B, are circularly equal, then there is a unique affine transformation $T : A \rightarrow B$. The area ratios that make up the canonical form are computed from overlapping point quadruples, which guarantees a unique affine transformation as follows. Given two equal canonical forms, the first corresponding quadruple of points defines a unique affine transformation (by Lemma 2). Each successive quadruple also defines a unique affine transformation, but that transformation shares three points with the transformation for the previous quadruple. Thus by Property 1 the successive transformations must be the same, because they share three corresponding points.

This completes the proof of Proposition 1, that the canonical forms of two point sets are circularly equal if and only if there is an affine transformation from one set to the other. Thus Algorithm 2 determines the existence of an affine transformation from A to B by comparing the canonical forms of two point sets.

Complexity

The time complexity of Algorithm 1 is $O(n \log n)$, as follows. The center of gravity can be computed in linear time. The next step involves determining a linear number of segments and angles, and then sorting this list, which takes time $O(n \log n)$. The third, fourth and fifth steps each compute a linear number of quantities, where computing each quantity is a constant time operation. Thus the sorting operation in the second step dominates the running time.

Algorithm 2 also requires $O(n \log n)$ time, because the first two steps each invoke Algorithm 1, and the third step can be done in linear time using one of a number of string matching algorithms. This time bound is optimal, by reduction from the problem of deciding whether two planar point sets are congruent, which is $\Omega(n \log n)$ [At].

Deciding whether two point sets are congruent can be reduced to the affine matching problem as follows. To decide if two point sets are congruent, first use Algorithm 2 to decide if there is an affine transformation from one set to the other. If there is no affine transformation then there is no congruence, as a congruence is a restricted form of affine transformation. If an affine transformation does exist, then use three corresponding points of the two sets (two corresponding segments in the sorted segment lists from Algorithm 1) to compute the transformation. Computing the transformation is a constant time operation, requiring the solution of two sets of three independent linear equations in three unknowns. If the matrix L of the affine transformation is orthonormal then the transformation is a congruence. If the matrix is not orthonormal then there is not a congruence. Determining whether the matrix is orthonormal is also a constant time operation.

4 Unequal Cardinality Point Sets

For any two sets of three points there always exists an affine transformation mapping one set to the other, but for sets of more than three points there generally is no such transformation. In this section we use four selected points of a set A to identify possible affine transformations from A to B, where $|A| < |B|$. Each transformation that maps the four chosen points of A to some four points of B is then checked by applying the transformation to the remaining points of A and using range queries to identify corresponding points in B. The method only considers pairs of points in B, rather than quadruples, by making use of the following result.

Proposition 2 *Given four points* $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$, *and four corresponding points* $\mathbf{a}'$, $\mathbf{b}'$, $\mathbf{c}'$, $\mathbf{d}'$, *where* $\mathbf{i}$ *is the intersection of* $\mathbf{ab}$ *with* $\mathbf{cd}$, *and* $\mathbf{i}'$ *is the intersection of* $\mathbf{a'b'}$ *with* $\mathbf{c'd'}$, *there exists a unique affine transformation mapping each of the four points to its corresponding point if and only if*

$$\frac{|\mathbf{ai}|}{|\mathbf{ab}|} = \frac{|\mathbf{a'i'}|}{|\mathbf{a'b'}|},$$

and

$$\frac{|\mathbf{ci}|}{|\mathbf{cd}|} = \frac{|\mathbf{c'i'}|}{|\mathbf{c'd'}|},$$

where $|x|$ *denotes the length of segment* x *(see Figure 2).*

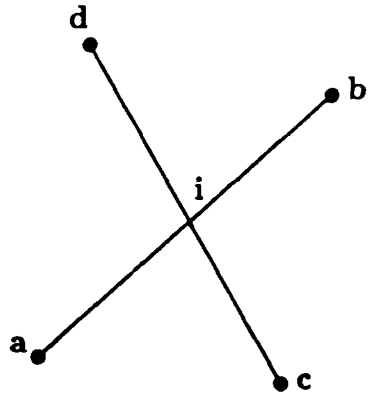

Figure 2: For a set of four points, the length ratios to the intersection point. **i** are preserved exactly when there is an affine transformation.

From Properties 2 and 3 it follows immediately that if the transformation from one set of points to the other is affine, then the length ratios to the intersection point will be preserved. The converse is also true, as follows. Choose three of the four points, such as **a**, **b**, **c** and **a**$'$, **b**$'$, **c**$'$. These three points always define a unique affine transformation, A (by Property 1). Now **i**$'$ must be the point along the line **a**$'$**b**$'$ that is specified the ratio **ai**/**ab**, and **d**$'$ must be the point along the line **c**$'$**i**$'$ that is specified by the ratio **ci**/**cd**. Thus **d**$'$ is defined uniquely given the affine transformation computed from the other three points, and the length ratios (that are invariant under an affine transformation).

Using this result, the affine transformations from a set of four points to a set of n points can be enumerated by considering only *pairs* of the n points. Given a set $A = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$, compute the intersection point, **i**, of the segment **ab** with the segment **cd**. For each pair of points $\mathbf{x}_i$ and $\mathbf{x}_j$ in B mark the four points along the segment $\mathbf{x}_i\mathbf{x}_j$ that are defined by the four length ratios $|\mathbf{ai}|/|\mathbf{ab}|$, $|\mathbf{ci}|/|\mathbf{cd}|$, and the *reverse* ratios $|\mathbf{bi}|/|\mathbf{ab}|$ and $|\mathbf{di}|/|\mathbf{cd}|$ (see Figure 3).

Two marked points will coincide, one from the ratio $|\mathbf{ai}|/|\mathbf{ab}|$ or its reverse and the other from the ratio $|\mathbf{ci}|/|\mathbf{cd}|$ or its reverse, when two edges intersect with the same length ratios as for the intersection of **ab** with **cd**. By Proposition 2 this happens if and only if the four points of B are an affine transformation of the four points of A. This basic idea can be generalized to an algorithm for matching m points to n points as follows.

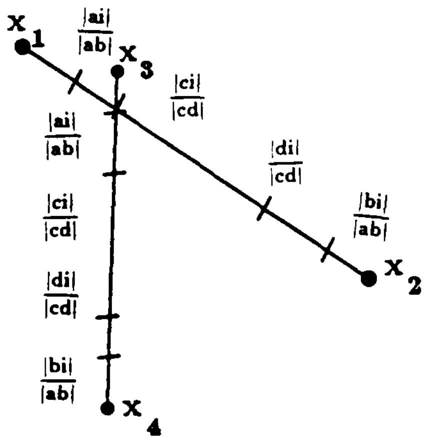

Figure 3: Two points specified by the length ratios coincide exactly when there is an affine transformation.

Algorithm 3 *Given two sets A and B of points in the plane, where $|A| < |B|$:*

1. *Choose a set of four points of A,* **a**, **b**, **c**, *and* **d** *such that the segments* **ab** *and* **cd** *intersect. Compute this intersection point,* **i**.

2. *Compute the length ratios* $|\mathbf{ai}|/|\mathbf{ab}|$ *and* $|\mathbf{ci}|/|\mathbf{cd}|$, *and the reverse ratios* $|\mathbf{bi}|/|\mathbf{ab}|$ *and* $|\mathbf{di}|/|\mathbf{cd}|$.

3. *For every pair of points,* $\mathbf{x}_i$ *and* $\mathbf{x}_j$ *in B, mark the four points along the segment* $\mathbf{x}_i\mathbf{x}_j$ *that are defined by the length ratios from Step 2. For each marked point note which segment of A (***ab** *or* **cd***) and which two points of B were used to compute it.*

4. *Collect all of the marked points from the previous step (along with which segment and which two points of B correspond to each point) into a list. Sort this list first by the x location of the marked points, then by the y location, and finally by whether the point resulted from an* **ab** *segment ratio or a* **cd** *segment ratio.*

5. *Equal marked points will be adjacent in the sorted list of the previous step, and will be separated into those corresponding to* **ab** *segments and those corresponding to* **cd** *segments. For each pair of equal points that is due to a different type of segment do the following:*

   (a) *Compute the affine transformation mapping the chosen points of A to the points of B that correspond to these marked points.*

(b) *Transform each point of A using the transformation computed in the previous step, and do a range query to determine if there is a matching point in B. If all the points of A have matches, then this transformation is valid and is collected into the result list.*

From Proposition 2 we know that a pair of equal points will be considered in Step 5 exactly when there is an affine transformation from the four chosen points of A to some four points of B. This transformation is then computed using the corresponding points, and is checked by transforming every point of A and determining whether there is a matching point of B. Thus every affine transformation of the m points of A to some m points of B will be exhibited.

Complexity

The first two steps of Algorithm 3 compute the intersection point and length ratios for the four chosen points of A, which takes a constant amount of time. The third step performs $O(n^2)$ constant time operations of marking points along a segment. The fourth step sorts a list of $O(n^2)$ points, and thus takes time $O(n^2 \log n)$ which dominates the running time of the first four steps.

The loop in step 5 is executed once for each pair of equal marked points, where one point is from an **ab** segment ratio and the other is from a **cd** segment ratio. Two such equal marked points will occur if and only if there is an affine transformation from the four chosen points of A to some four points of B. Thus there can be at most $O(n^3)$ iterations of the loop. In general, however, the number of iterations will be small, because the number of affine matches of four points is small. Each time around the loop takes $O(m)$ time to transform the points of A, and then $O(m \log n)$ to perform m range queries on the n points of B. Thus the overall running time of the algorithm is $O(n^2 \log n + tm \log n)$, where t is the number of transformations from the four chosen points of A to any four points of B, and hence in the worst case t is $O(n^3)$.

5 Summary

We have considered the problem of determining whether one set of points in the plane matches another set under an affine transformation. We presented an algorithm for the case where the two points sets are of equal cardinality that decides whether there is an affine transformation in optimal time $\Theta(n \log n)$. When a set

A of m points is matched against a set B of n points, where $m < n$, we have developed a method that exhibits all the affine transformations from A to B (if any) and runs in time $O(n^2 \log n + tm \log n)$, where t is the number of matches of a given four points in A to any four points in B. In general t is small, but in the worst case it is $O(n^3)$.

An interesting related problem is matching under an affine transformation when there is bounded uncertainty in the locations of the points. If the transformation is restricted to be a congruence, rather than a general affine transformation, and the cardinality of the two point sets is the same, then an $O(n^8)$ algorithm for approximate matching is given in [AMWW]. The problems of approximate matching under an affine transformation, and of matching unequal cardinality point sets remain open.

In this paper we have only considered the unequal cardinality matching problem in which each point of a set A is matched to some point of B. In real-world machine vision tasks, however, some of the points of A may not be present in B. Thus, another related question is how to match A to B when there may not only be "extraneous" points in B, but may also be points of A that are missing from B.

Acknowledgments

The authors would like to thank Peter Wayner for his comments on a draft of this paper.

References

[AMWW] Alt, H., Mehlhorn, K., Wagener, H. and Welzl, E. "Congruence, Similarity, and Symmetries of Geometric Objects", *Discrete and Computational Geometry*, vol. 3, 237-256, 1988.

[At] Atallah, M.J. "Similarity of Planar Figures" *Int. J. Comput. Inf. Sci.*, vol. 13, 279-290, 1984.

[Ba] Baird, H.S. *Model-Based Image Matching Using Location*, MIT Press, 1985.

[Ed] Edelsbrunner, H. *Algorithms in Combinatorial Geometry*, Springer-Verlag, 1987.

[Ga] Gans, D. *Transformations and Geometries*, Appleton-Century-Crofts. New York, 1969.

[HT] Hong, J. and Tan, X. "Recognize the Similarity Between Shapes under Affine Transformation", *Proc. Second Intl. Conf. on Computer Vision*, IEEE Computer Society Press, 1988.

[Ho] Horn, B.K.P. *Robot Vision*, MIT Press, Cambridge, Massachussets, 1986.

[HU] Huttenlocher, D.P. and Ullman, S. "Object Recognition Using Alignment", *Proc. First Intl. Conf. on Computer Vision*, pp. 102-111, IEEE Computer Society Press, 1987.

[Kl] Klein, F. *Elementary Mathematics from an Advanced Standpoint: Geometry*, MacMillan, New York, 1939.

[LSW] Lamdan, Y., Schwartz, J.T. and Wolfson, H.J. "Object Recognition by Affine Invariant Matching", *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE Computer Society Press, 1988.

[TM] Thompson, D. and Mundy, J.L., "Three-Dimensional Model Matching From an Unconstrained Viewpoint", *Proc. IEEE Conf. Robotics and Automation*, IEEE Computer Society Press, 1987.