A Continuation Method for Pose Estimation and Correspondence

Suguna Pappu, Steven Gold and Anand Rangarajan

Research Report YALEU/DCS/RR-1061
January 1995

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

A Continuation Method for Pose Estimation and Correspondence

Suguna Pappu, Steven Gold, and Anand Rangarajan[1]
Yale Center for Theoretical and Applied Neuroscience (CTAN)
and Department of Computer Science
Yale University

Abstract

Matching feature point sets lies at the core of many approaches to object recognition. This entails estimating the pose and finding the correspondence between the sets. A feature matching approach is presented using a match matrix to represent correspondence, with the pose an affine transformation. The problem is formulated as an optimization problem for determining the correspondence and pose between feature points in R^n, constrained by the values of the match variables and the class of affine mappings. A continuation method is developed for the special case of point matching, and then extended to incorporate multiple types of feature information. An iterative algorithm is obtained that alternates between match matrix constraint satisfaction via a projective scaling method and closed form update of the affine mapping. The strength of the algorithm is its ability to handle noisy data sets, unequal in size. Results for synthetic and real world data are provided for point sets in 2D and 3D, and for 2D data with multiple types of features.

Keywords: Object recognition, feature matching, point matching, affine transformation, correspondence

[1]e-mail address of authors: lastname-firstname@cs.yale.edu

1 Introduction

A basic problem of object recognition is that of matching– how to associate sensory data with the representation of a known object. In deformable template matching the features of an unknown object in an image are given as input information and the objective is to match these features with those of a known object model. The features chosen for a representation are typically context-driven (Grimson, 1990; Faugeras, 1994). Spatial location of the components of the object is important, complemented by geometric information, such as whether a point in the space is on a curve or is a breakpoint etc., and non-geometric features like color and texture. However, the data garnered by the feature extraction may be noisy or incomplete, due to limitations of image processing, or because of occlusion or viewing geometry.

We present a method for feature matching, in which the primitives of a certain shape representation are given, e.g., points, lines, curvature, corners, texture. The matching of one object to another includes the *correspondence* of similar parts of the two objects, and determination of the correct *pose*, the accurate mapping of the model onto the image. The approach we will detail is robust with respect to noise and incomplete information. This multiple feature matching scheme can be seen as a module within a general purpose object recognition system. It may interact with other modules, such as one that classifies structure based on attribute relationships and then be input to a higher level classification module.

We detail the elements of our approach by considering affine point matching as a special case of the general feature matching problem. The set of point features can be an effective representation for many problems (Grimson, 1990; Umeyama, 1993). For us, point features serve as a base description in a multi-feature model. However, inclusion of other feature information can improve the quality of the matching found, particularly in the case of noise and occlusion.

The point matching problem is formulated as finding the correspondence between two point sets, perhaps of unequal size, and determining the correct pose transformation. In this paper, we examine the class of affine mappings, but we indicate how it can be extended to more general problems. The correspondence between a point on the image and a point on the model is made explicit via a match variable. The matrix of match variables indicates the degree of correspondence between the two sets. We use a distance measure that is a function of the sum of the distances between all of the points in correspondence, in turn a function of the affine mapping. We minimize this distance measure via a continuation method (Blake and Zisserman, 1987; Geiger and Girosi, 1991; Leclerc, 1989).

Our formulation is designed for multi-feature template matching, for which point matching is a special case. Although most point matching algorithms advertise themselves as *feature* matching schemes, there has been very little work on the integration of different types of feature information to establish correspondence between objects. The primary strengths of our algorithm are its ability to incorporate high level features, and to operate in significant noise. We use both synthetic and real-world examples to illustrate these aspects.

2 Previous Work

The point matching problem has been approached in several ways. These may be categorized as tree-pruning methods, voting schemes, eigenvectors approaches, relaxation labelling algorithms and neural network methods. A tree-pruning method is a systematic search over the set of possible matches, with portions of the search space eliminated due to inconsistencies or constraint violation (Baird, 1984; Grimson and Lozano-Perez, 1987; Umeyama, 1993). The generalized Hough transform is a voting scheme which operates in the parameter space of the transformation, dividing the space into discrete bins, each of which receives a vote when a good match is found corresponding to that bin (Ballard, 1981; Stockman, 1987). Geometric hashing is a voting scheme that operates in the point space (Lamdan et al., 1988; Hummel and Wolfson, 1988; Costa et al., 1990; Ogawa, 1986). (Scott and Longuet-Higgins, 1991; Shapiro and Brady, 1992; Sclaroff and Pentland, 1993) all propose eigenvector based approaches, which we describe below. Relaxation methods are used by (Ranade and Rosenfeld, 1980; Ton and Jain, 1989). (Vinod and Ghose, 1993; Gee et al., 1993) have developed neural net schemes.

(Scott and Longuet-Higgins, 1991) have a formulation for point matching that is similar to ours, with a pairing matrix that indicates the matches and a proximity matrix whose elements are the pairwise distances between points. The proximity matrix is a function of a Gaussian weighted distance metric, with a parameter σ which controls the degree of interaction between the two sets of features. Their eigenvector based approach computes the modes of the proximity matrix, and they show that for a value of σ large enough, they recover the correct global correspondence. (Shapiro and Brady, 1992) continue this work by including modal shape information to address the weaknesses of Scott and Longuet-Higgins's algorithm, primarily its inability to recover large rotations in the transformations, and also that the assumption of large σ may result in algorithmic instabilities. A more recent approach by (Manmatha, 1994) handles lines and images as well as points. However, none of these approaches have an explicit formulation that requires the one-to-one matching of points.

Precursors of our work include (Yuille, 1990; Hinton et al., 1992; Lu and Mjolsness, 1994). However, each tells only part of the story. Specifically, none of them have provided a formulation that integrates the key elements of establishing a one-to-one correspondence between point sets, related by some type of non-rigid mapping, and where the sets may be of unequal size or characterized by noise. Addressing all of these points is a crucial step in the development of a viable object recognition system. We have succeeded in doing this not only for point matching but also for the more general multi-feature matching problem. Specifically, the constraints on the match matrix ensure the one-to-one mapping, and the use of slack variables permits matching data sets of unequal size. The class of affine mappings is a tractable set of non-rigid transformations. We then provide an efficient algorithm that works well in this context. In (Gold et al., 1994) we first introduced several of these elements for point matching. In particular, the iterative projective scaling algorithm used here to satisfy the match matrix constraints was first detailed, although unequal point sets and non-rigid transformations were not included. (Gold et al., 1995) extended this work to account for unequal 2D point sets and affine transformations, with the individual terms of the transformation, translation, rotation, scale and oblique and vertical shears, optimized in coordinate-wise fashion. 3D results for rigid transformations only, on unequal point sets were presented using dual number quaternions to compute the transformation parameters.

In this current work, the affine transformation is solved in closed form. This enables us to extend our previous results to affine transformations on 3D point sets. We then show that the point matching algorithm is a special case of a multi-feature matching scheme. Finally, we sketch how the closed form solution also permits straightforward extensions to higher-order polynomial transformations and data sets in any dimension, without modification of the basic methodology.

3 Affine Point Matching

The algorithm has two components– i) *Correspondence*: determining which point in the image set matches to which point (if any) in the model set, and ii) *Pose estimation*: computing the affine transformation that maps the model points of the correspondence onto the image set, minimizing a distance measure. There are several instances in which an affine mapping is appropriate, such as in weak perspective projection. In many cases the affine mapping may be sufficient to recover the correct correspondence even when the transformation itself is clearly not affine. We will demonstrate several examples of this.

3.1 Problem Formulation

Given two sets of data points $\hat{X}_j \in R^{n-1}$, $n = 3, 4 \ldots,$ $j = 1, \ldots, J$, the image, and $\hat{Y}_k \in R^{n-1}$, $n = 3, 4, \ldots, k = 1, \ldots, K$, the model, the problem is to find the correspondence between the two sets and associated affine transformation that best maps a subset of the image points onto a subset of the model point set. The best mapping is determined by minimizing a distance measure that is a function of the model points as matched to the transformed image points.

The class of affine mappings is the most tractable of the non-rigid transformations. It includes translation, rotation, scale, and both horizontal and oblique shears. In the case of point features, the squared Euclidean norm is an appropriate distance measure between model and images points. These point sets are expressed in homogeneous coordinates, $X_j = (1, \hat{X}_j)$, $Y_k = (1, \hat{Y}_k)$. Let $A \in R^{n \times n}$ be the affine transformation matrix. The use of homogeneous coordinates permits the translation component of the transformation to be concisely included in the matrix A, given by the first column. Additionally, the elements of the first row of A are equal to zero because of the homogeneous coordinate representation. The distance between the model point X_j and the transformed Y_k is given by

$$\|X_j - (A + I)Y_k\|^2 \tag{1}$$

where I is the identity matrix of dimension n.

The explicit association of the set of model points either with a point in the image set or a null point, indicating that it does not match to any image point, is expressed via a matrix of match variables, M_{jk} where

$$0 \leq M_{jk} \leq 1 \tag{2}$$

$M_{jk} = 1$ indicates that X_j is fully assigned to Y_k. $M_{jk} \in (0,1)$ is a *partial* assignment of X_j to Y_k. For a given match matrix $\{M_{jk}\}$ and transformation A,

$$\sum_{j,k} M_{jk} \| X_j - (A+I)Y_k \|^2 \tag{3}$$

expresses the similarity between point sets that we are attempting to match.

However, this expression (3) needs to be modified and constrained in several ways to reflect physical constraints. First, the total assignment of each variable is at most 1, i.e., X_j may fully match to one Y_k, partially match to several upto a total value of 1, or may not match to any point. A similar constraint holds for Y_k. In this way the constraint on M_{jk} becomes

$$\begin{aligned} \sum_j M_{jk} &\leq 1, \quad \forall k \\ \sum_k M_{jk} &\leq 1, \quad \forall j \\ \text{and } M_{jk} &\geq 0 \end{aligned} \tag{4}$$

Note that the special case in which the number of points in each data set is equal and M is constrained by an integrality requirement, the feasible set is the space of permutation matrices in $R^{n \times n}$.

Second, the distance measure itself needs to be modified in order to ensure reasonable solutions. When (3) is minimized subject to the constraints in (4) we obtain $M_{jk} = 0$ for all values.

To counter this, we append a term $-\alpha \sum_{j,k} M_{jk}$, with parameter $\alpha > 0$ that encourages matches. Additionally, we are interested in small to medium range affine transformations, so the terms of the affine matrix are regularized. This is represented by a term $\lambda tr(A^T A)$ in the objective function, with parameter λ, and $tr(.)$ denoting the trace of the matrix.

The correspondence and pose estimation problem is to minimize the objective function (5) subject to the constraints on the correspondence matrix.

$$\min_{A,M} \sum_{j,k} M_{jk} \|X_j - (A+I)Y_k\|^2 + \lambda tr(A^T A) - \alpha \sum_{j,k} M_{jk} \tag{5}$$

s.t.

$$\sum_j M_{jk} \leq 1, \quad \forall k$$
$$\sum_k M_{jk} \leq 1, \quad \forall j$$
$$M_{jk} \geq 0$$

Note that because M appears linearly in the objective function, the simplex constraints on M indicate that any local minimum of (5) occurs at a vertex of the hypercube of feasible solutions. In our algorithm we will design a trajectory for M that is on the interior of this hypercube and will avoid some local minima.

3.2 Algorithm Description

The algorithm follows by taking the given constrained optimization problem in (5) and deriving an equivalent unconstrained problem. The inequality constraints (4) can be transformed into equality constraints, introducing slack variables $M_{j,K+1}$ and $M_{J+1,k}$ so that

$$\sum_{j=1}^{J} M_{jk} \leq 1 \rightarrow \sum_{j=1}^{J+1} M_{jk} = 1, \quad \forall k$$
$$\sum_{k=1}^{K} M_{jk} \leq 1 \rightarrow \sum_{k=1}^{K+1} M_{jk} = 1, \quad \forall j$$

$M_{j,K+1} = 1$ indicates that X_j does not match to any point in Y_k. The match matrix is depicted in Figure (1).

We derive an equivalent optimization problem to (5) by relaxing these constraints via Lagrange parameters μ_j, ν_k, and introducing an $x \log x$ barrier function, indexed by a parameter β. The barrier function enforces the non-negativity of the match variables. This function was used by (Kosowsky and Yuille, 1994) to solve the well-known combinatorial optimization problem, the assignment problem, via a continuation method. Their approach was not applied to point matching and they did not consider the case of unequal numbers of points.

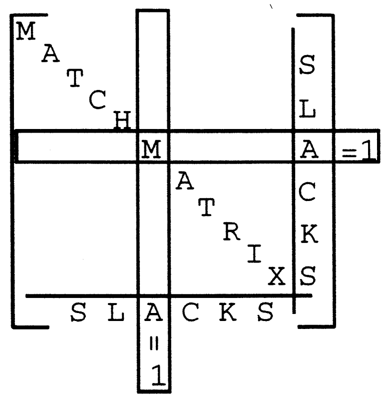


Figure 1: Match Matrix, including slack variables

The new optimization problem is:

$$\min_{A,M} \max_{\mu,\nu} \sum_{j,k} M_{jk} \|X_j - (A+I)Y_k\|^2 + \lambda tr(A^T A) - \alpha \sum_{j,k} M_{jk} + \sum_{j}^{J} \mu_j (\sum_{k=1}^{K+1} M_{jk} - 1) \tag{6}$$

$$+ \sum_{k}^{K} \nu_k (\sum_{j=1}^{J+1} M_{jk} - 1) + \frac{1}{\beta} \sum_{j=1}^{J} \sum_{k=1}^{K} M_{jk} (\log M_{jk} - 1)$$

As with most constrained non-linear optimization problems, we are going to be minimizing with respect to the match variables and affine transformation parameters while satisfying the constraints via Lagrange parameters.

The algorithm alternately updates the match matrix and the affine transform. Suppose that the correspondence between the two point sets is known, i.e. M is fixed. In this case, we can solve in closed form for the affine matrix. Let $f(A, M)$ denote the value of the objective function. Solving for A we get,

$$\frac{\delta f(A, M)}{\delta A} = \sum_{j,k} M_{jk} ((A+I)Y_k - X_j) Y_k^T + \lambda A = 0 \tag{7}$$

$$A = A^*(M) = (\sum_{j,k} M_{jk} (X_j Y_k^T - Y_k Y_k^T))(\sum_{j,k} M_{jk} Y_k Y_k^T + \lambda I)^{-1} \tag{8}$$

When A is known, we are effectively solving an assignment problem. Popular discrete methods for solving it are the Hungarian method (Papadimitriou and Steiglitz, 1982) and the auction algorithm (Bertsekas, 1981) As previously mentioned, the continuation method of (Kosowsky and Yuille, 1994) is based on the minimization of a related continuous non-linear optimization problem. In our technique, the assignment constraints are satisfied by a continuation method, with continuation parameter $\beta \to \infty$, associated with the barrier function (Blake and Zisserman, 1987; Geiger and Girosi, 1991; Peterson and Soderberg, 1989). We use the result of (Sinkhorn, 1964) who shows that repeated row-column normalization of a doubly stochastic matrix converges to a permutation matrix. From this we develop an iterative projective scaling procedure detailed in (Gold et al., 1994) to find the correct assignment by iteratively doing row-column normalization on the match matrix to satisfy the assignment constraints.

The value of the match variable is initialized to be a function of the continuation parameter β and the current affine matrix A. Specifically, we use the solution to (6) when minimized with respect to M_{jk}, and without the row and column constraints.

$$M^*_{jk}(A) = \exp(-\beta \|X_j - (I + A)Y_k\|^2 - \alpha) \tag{9}$$

This is followed by a repeated row-column normalization, iterative projective scaling, of the match variables until a stopping criterion is reached:

$$M_{jk} = \frac{M_{jk}}{\sum_{j'} M_{jk}} \text{ then } M_{jk} = \frac{M_{jk}}{\sum_{k'} M_{jk}} \tag{10}$$

Finally, A is set using equation (8). These steps constitute the basic elements of the algorithm, with β following a fixed schedule, and the updates of A and M as described.

Pseudocode: Point Matching

1. INITIALIZE: Fixed β, do T times:

   Parameters: $\beta_{\text{initial}}, \beta_{\text{update}}, \beta_{\text{final}}$ $T =$ Iteration number, λ

   Variables: $A = 0$, $M = 0$

2. ITERATE: Fixed β, do T times:

   Re-initialize $M^*(A)$ (Eq. 9)

   Row-column normalization until ΔM small

   $A^*(M)$ updated (Eq. 8)

3. UPDATE: While $\beta < \beta_{\text{final}}$

   $\beta \leftarrow \beta * \beta_{\text{update}}$

   Return to 2.

This is depicted in Figure (2). The complexity of the algorithm is $O(JK)$. Note that the computation of the Lagrange parameter values is unnecessary because of the explicit satisfaction of the constraints. Starting with small β_{initial} permits many partial correspondences

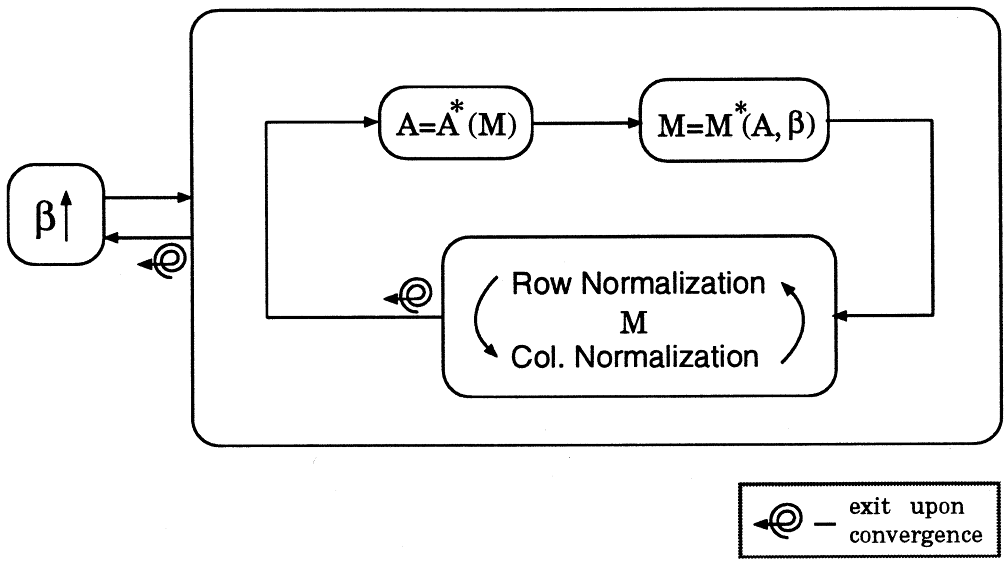


Figure 2: Flowchart of the Algorithm

in the initial solution for M. As β increases the correspondence becomes more refined. A value of β_{final} that is large will yield solutions for M that approach a binary matrix.

4 Multiple Feature Matching

It is clear that the recognition of an object requires many different types of information working in concert. Additionally, the information itself may be noisy or unclear. These factors motivate the development of a multi-feature model, and the formulation we have developed so far is well-suited for this extension. The optimization problem formulated in (5) for point matching can be generalized in a straightforward way to incorporate multiple features to determine correspondence. A representation with multiple features has a spatial component which indicates the location of a feature element, i.e., the point coordinates. At that location, then, there may be other invariant geometric characteristics, such as the identification of a point on a curve or at a breakpoint. Examples of non-geometric invariant features are color, curvature and texture.

Extending the notation to explicitly include the different features, X_{jr} is the value of feature r associated with point X_j. The *location* of point X_j is the null feature. In this way, there are R features associated with each point X_j and Y_k. The match variable remains the same. The key notion that we are trying to capture in a multi-feature model is that many different types of information associated with a data element may interact to determine the correct match. The new objective function is

$$\sum_{j,k} M_{jk}\|X_j - (A+I)Y_k\|^2 + \sum_{j,k,r} M_{jk}w_r(X_{jr} - Y_{kr})^2 + \lambda tr(A^T A) - \alpha \sum_{j,k} M_{jk}$$

$(X_{jr} - Y_{kr})^2$ is a term that captures the similarity between invariant types of features, with w_r a weighting factor for feature r. In the case that the feature is not invariant, a different term must be chosen.

The algorithm as described for the point matching problem is modified only in the re-initialization of $M(A)$. This term becomes

$$M_{jk} = \exp(-\beta(\|X_j - (I+A)Y_k\|^2 + \sum_r w_r(X_{jr} - Y_{kr})^2 - \alpha)) \tag{11}$$

The rest of the algorithm remains unchanged.

5 Experimental Results

The speed for matching point sets of 50 points each is around 20 seconds on a Silicon Graphics workstation with a R4400 processor. This is true for points in 2D, 3D and with extra features. This can be improved with a tradeoff in accuracy by adopting a looser schedule for the continuation parameter β or by changing the stopping criterion. We have done several types of experiments to illustrate the strengths of our algorithm. We have generated several line drawings to demonstrate the robustness of the correspondence, even for significant changes in scale, distortions etc. Synthetic results for point sets in 2D and 3D show that the recovery of the correspondence and the affine transformation is good, even in the presence of significant noise and point addition/deletions and for a large range of affine transformations.

5.1 Hand Drawn Examples

The data were generated using an X-windows tool which enables us to draw an image with the mouse which appears on the screen. The contours of the images are discretized and are expressed as a set of points in the plane. The point sets for all the following 2D examples in this section were generated in this manner.

In the first experiment, we drew alphanumeric characters and words. In the experiments below, we generated 70 points per character on average. The inputs to the point matching algorithm are the x-y coordinates generated by the drawing program. No other pre-processing is done. The output is a correspondence matrix and a pose. In Figures (3) and (4), we show the correspondences found between several images drawn in this fashion. To make the actual point matches easier to see, we have drawn the correspondences only for a subset of the model points. In one experiment, we drew examples of individual digits, one as a model digit and

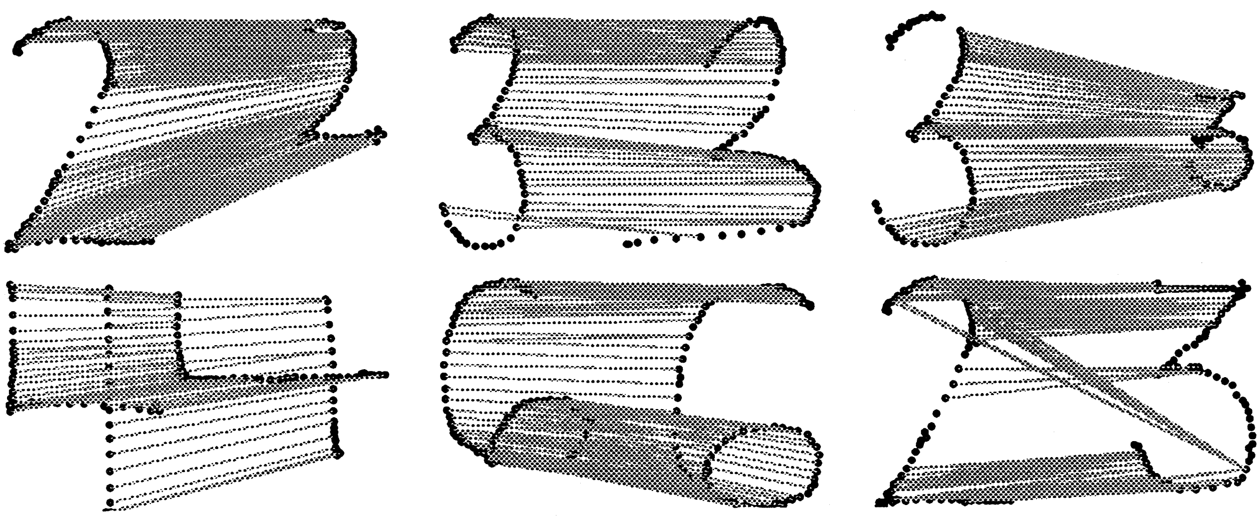


Figure 3: Digit Correspondences

then many different variations of it. In Figure (3) it can be seen that the correspondences are good for a large variation from the model digit. In particular, note that the correspondence is quite invariant to scale, as in the first and third examples in the top row. We are able to handle large distortions, including both scale and shear as seen in several examples. The poor correspondence found in matching two different digits is vividly depicted in the lower right corner.

In another experiment, we were interested in identifying an individual letter correctly within a word, both of which are handwritten in cursive form. An example is shown in Figure (4), Note that the correspondence itself is good and the issue of segmentation of a cursive word is not addressed separately, nor is there any other basis for identification other than the point matching algorithm itself. In particular, note that the "o" is correctly identified in "song," not with the similar looking "s," and despite the slight overlap in the letters.

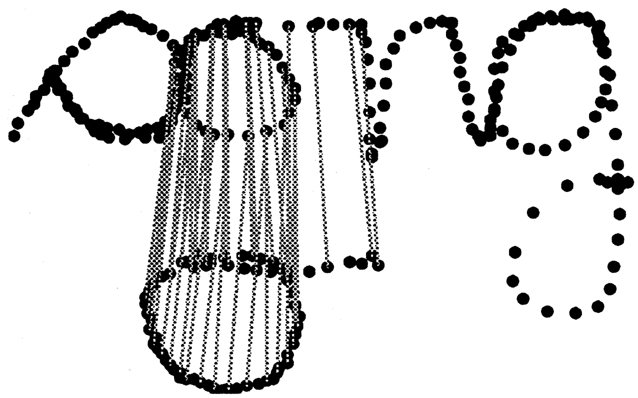


Figure 4: Correspondence: "o" in *song*

Figure 5: Features of the face match

Next, we drew pictures with distinct parts, shown in Figure (5). The contours of the boy's face were drawn in two different positions, and a subset of the points were extracted to make up the point sets. In each set this was approximately 250 points. Note that even with the change in mood from one picture to another the correct correspondence is found for the individual parts of the face. These results using only point features are promising.

For handwritten characters, these results can be improved by including feature types specific to handwriting. Stroke analysis indicates that different types of inflection points may be such features (Edelman et al., 1990). Similarly, the parts of the boy's face may have certain feature markers.

With this in mind, to illustrate our work using a multi-feature model, we use pictures of a person taken at slightly different angles, shown in Figure (6). Although the rotation of the head is not a true affine transformation, it may be seen as a weak perspective projection for which an affine approximation is valid. Each photo is outlined using a graphics drawing tool. The contours are discretized and a sample of the point set is taken, approximately 225 points in each face.

Figure 6: Original images

A point on a contour has associated with it a feature marker indicating the incident textures. For a human face, we use a binary 4-vector, with a 1 in position r if feature r is present. Specifically, we have used a vector with elements [*skin, hair, lip, eye*] For example, the lines marking the mouth segment the lip texture from the skin texture. A point within this line has a feature vector $[1,0,1,0]$. The nose has only adjacent skin and so a point on one of these lines has a feature vector $[1,0,0,0]$. Perceptual organization of the face justifies this type of feature marking scheme. Figure (6) was marked in this manner with the

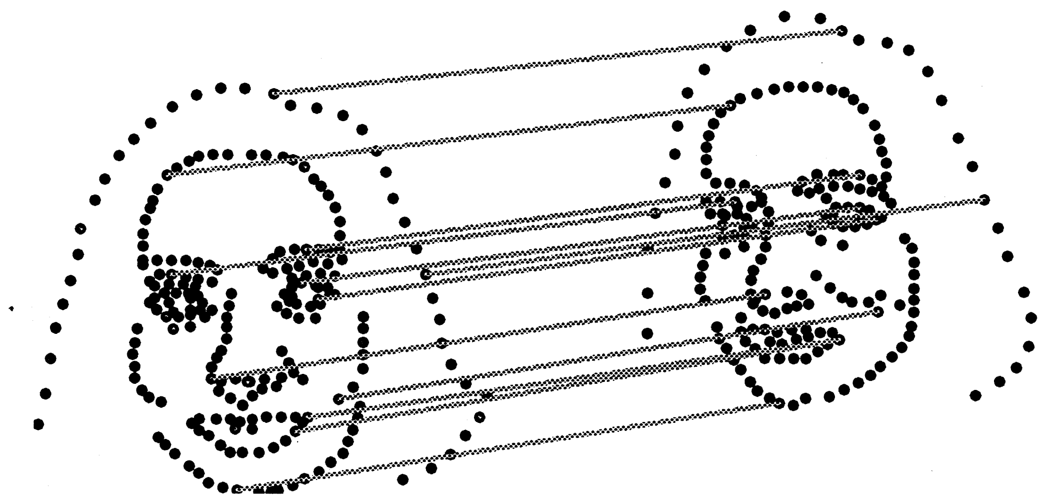


Figure 7: Correspondence with labelled features

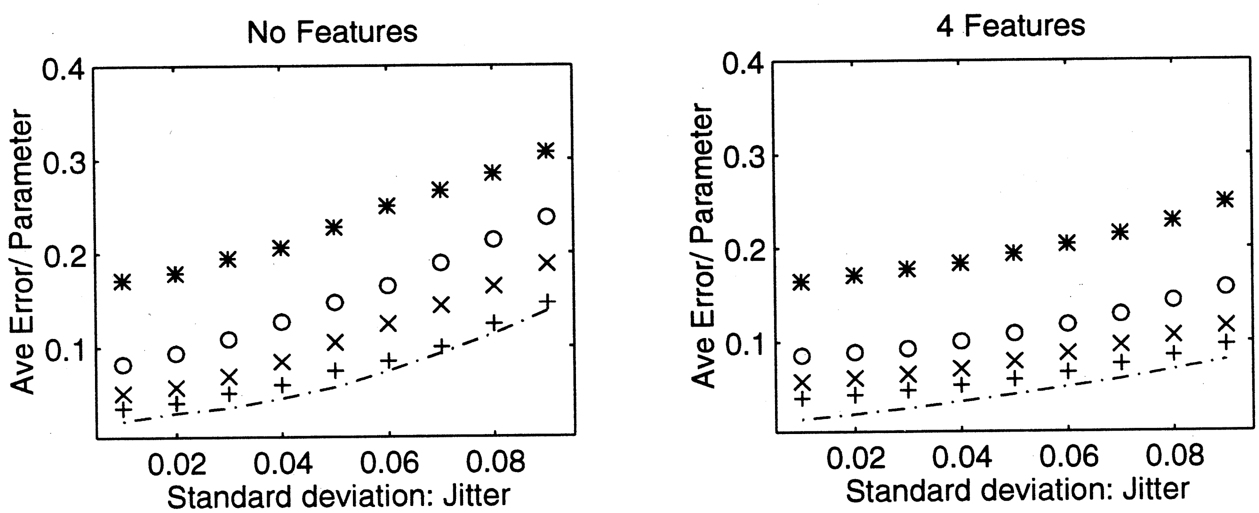
resulting correspondence depicted in Figure (7). We have shown only a small subset of the matches in order to see the results more clearly. Note that there are several areas in which the correspondence could be easily thrown off, e.g., in the eye region the eye could easily have matched to a point in the dense neighborhoods of eyebrow or nose.

5.2 Synthetic Data

Experiments on synthetic data were performed to test the strength of the algorithm in the presence of noise and missing or extra points. These experiments are designed to show how, when the transformation is indeed affine, that the correct mapping is recovered.

The model set was generated uniformly on a unit square. A random affine matrix is generated, where each of the affine parameters, a_{ij} are chosen uniformly on a certain interval, and applied to the model set to generate the image set. Points may be deleted or spurious points may be included, and the points may be "jittered" by some Gaussian noise. This point set constitutes the test image from which we seek to recover the original affine transformation. In the case that other types of feature information are available, a binary valued feature vector is randomly chosen.

We show the results of experiments structured in this way for 2D data in Figure (8). 50 model points were generated, and the affine parameters were chosen uniformly on an interval of length 1.5 units. This affine mapping is then applied to the data set. A subset of these image points p_d are then deleted, and then Gaussian noise, $N(0, \sigma)$ is added. Finally, spurious points, p_s are added. The elements of the feature vector are randomly mislabelled with probability, P_r, to represent a distortion of the features. In our experiments, we have chosen $p_d \in \{0\%, 10\%, 30\%, 50\%, 70\%\}$ and $\sigma \in \{0.01, 0.02, \ldots, 0.08\}$. Spurious points are added only when there are point deletions, with $p_s = 10\%$ for $p_d > 0$. For experiments using multiple features, the random feature noise has a probability of $P_r = 0.05$.



$-. : p_d = 0.0, p_s = 0.0,$ $\quad$ $o : p_d = 0.50, p_s = 0.1$

$+ : p_d = 0.10, p_s = 0.1,$ $\quad$ $* : p_d = 0.75, p_s = 0.1$

$x : p_d = 0.30, p_s = 0.0$

Figure 8: 2D results for synthetic data

The error measure we use is

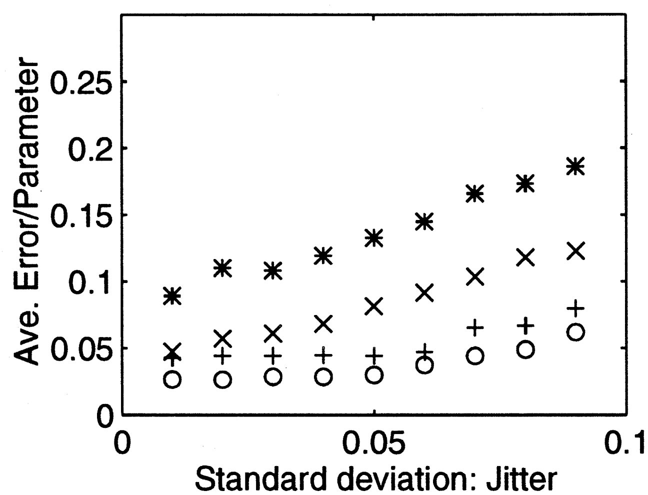
$$e_a = c \sum_{i,j} |a_{ij} - \hat{a}_{ij}| \tag{12}$$

where a_{ij} is the correct parameter, and $\hat{a}_{ij}$ is the computed value. The constant term c normalizes the measure so that the error equals 1 in the case that the a_{ij} and $\hat{a}_{ij}$ are chosen at random on this interval. Specifically, this is derived by

$$c = \frac{3}{\# \text{ of parameters}} \frac{1}{\text{length of interval}} \tag{13}$$

The factor 3 in the numerator of this formula follows because the expected value of the absolute difference of two numbers chosen randomly on the unit interval is $\frac{1}{3}$, and this value normalizes the measure to 1. Thus, $\frac{1}{3} = \frac{3}{6}\frac{1}{1.5}$. This is the error measure reported in Figure (8). Each data point represents 500 runs for a different randomly generated affine transformation. The parameters used were: $\beta_{\text{initial}} = .091$, $\beta_{\text{final}} = 100$, $\beta_{\text{update}} = 1.075$, and $T = 4$. For the experiments with features, we used $R = 4$ features, and $w_r = 0.2$, $\forall r$.

As is expected, the inclusion of feature information reduces the error. This is particularly evident for large amounts of jitter. Also, note from Figure (8) that for small quantities of point additions and deletions, the error is not substantially more than that for the base case in which there are no additions or deletions.



o : $p_d = 0.0, p_s = 0.0,$ + : $p_d = 0.10, p_s = 0.1$

x : $p_d = 0.30, p_s = 0.1,$ $*$: $p_d = 0.50, p_s = 0.1$

Figure 9: 3D results for synthetic data

In Figure (9) we performed similar experiments for point sets that are 3-dimensional, but without any feature information. As in the 2D case above, experiments were performed, with $p_d \in \{0\%, 10\%, 30\%, 50\%\}$ and $\sigma \in \{0.01, 0.02, \ldots, 0.08\}$. Spurious points are added only when there are point deletions, with $p_s = 10\%$ for $p_d > 0$.

The error measure is $\frac{1}{6}\sum_{i,j} |a_{ij} - \hat{a}_{ij}|$, with the factor $\frac{1}{6}$ used to normalize the 12 parameters of the transformation over the interval of length 1.5 units, derived from the formula (13).

6 Discussion

We have presented a method for feature matching that is effective in many contexts, particularly because of its robustness in the presence of noise. Also, the overall simplicity of the algorithm permits its use as a skeleton module easily incorporated in a larger scheme. Here, we indicate several extensions that are immediate. The non-binary valued match matrix already indicates the strength of a match as opposed to an absolute assignment. In the case that the feature information is not known with certainty, weights reflecting our confidence in each point feature can be included in the matching.

Note that the transformation described here is not restricted to the familiar first-order affine mapping; higher order polynomial transformations may be of use in the case of local warpings. In the case that a second order transformation is sought, the vector X_j is given by $[1, X_{j1}, X_{j2}, X_{j1}^2, X_{j2}^2, X_{j1}X_{j2}]$, and the transformation matrix requires recovery of 30 parameters. Structurally, the algorithm remains the same. Alternatively, one can think of a quilt of local affine mappings, in which the space is subdivided into many regions, each region subject to a different mapping. Additionally, since the class of affine mappings is a tractable set of non-rigid transformations, and serves as a first-order approximation of the pose, it may be used as an initial estimate in a coarse-to-fine type of hierarchy. Or, a multi-feature scheme may permit the affine to effect a good first order change, with the remainder of the correspondence derived from the other feature information.

The algorithm itself is inherently parallel, in terms of the operations on the match variables, and this can be exploited for any application that must be done in real-time. Specifically, since the operations on the match matrix give the algorithm a complexity of $O(n^2)$, parallelizing these operations over n-processors results in a linear time implementation. A machine whose processors are configured for single-instruction, multiple data (SIMD) execution has the potential to permit such a fast implementation.

We have presented a continuation method that integrates multiple types of feature information to determine correspondence between data sets, related spatially by an affine mapping. Noisy, incomplete feature information limits the applicability of many matching schemes. However, our easily implementable algorithm is effective in these environments, especially when there is non-spatial feature information to supplement the point location data. The class of first order affine transformations that we have considered may be generalized in several ways as we have indicated, for a higher order transformation or to be used as a coarse approximation. In these ways, our model is applicable for more complex non-rigid transformations without any modification in the methodology. The general applicability and straightforward extensibility of our model support its use as a deformable template matching module in an object recognition system.

Acknowledgements

This work was supported by the Yale Center for Theoretical and Applied Neuroscience (CTAN) and ONR/DARPA N00014-92-J-4048. We thank Eric Mjolsness and Chien-Ping Lu for many useful discussions, and Manisha Ranade for technical support.

References

Baird, H. (1984). *Model-Based Image Matching Using Location.* MIT Press, Cambridge, MA.

Ballard, D. (1981). Generalized Hough transform to detect arbitrary patterns. *Pattern Recognition*, 13(2):111–122.

Bertsekas, D. (1981). A new algorithm for the assignment problem. *Mathematical Programming*, 21:152–171.

Blake, A. and Zisserman, A. (1987). *Visual Reconstruction.* MIT Press, Cambridge, MA.

Costa, M., Haralick, R., and Shapiro, L. (1990). Optimal affine-invariant point matching. In *10th ICPR*, pages 233–236. IEEE Press.

Edelman, S., Flash, T., and Ullman, S. (1990). Reading cursive handwriting by alignment of letter prototypes. *International Journal of Computer Vision*, 5(3):303–33.

Faugeras, O. (1994). *Three dimensional Computer Vision.* MIT Press, Cambridge, MA.

Gee, A., Aiyer, S., and Prager, R. (1993). An analytical framework for optimizing neural networks. *Neural Networks*, 6:79–97.

Geiger, D. and Girosi, F. (1991). Parallel and deterministic algorithms from MRFs: Surface reconstruction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):401–412.

Gold, S., Lu, C. P., Rangarajan, A., Pappu, S., and Mjolsness, E. (1995). New algorithms for 2D and 3D point matching: Pose estimation and correspondence. In Tesauro, G., Touretzky, D., and Alspector, J., editors, *Advances in Neural Information Processing Systems*, volume 7, San Francisco, CA. Morgan Kaufmann Publishers.

Gold, S., Mjolsness, E., and Rangarajan, A. (1994). Clustering with a domain-specific distance measure. In Cowan, J. D., Tesauro, G., and Alspector, J., editors, *Advances in Neural Information Processing Systems*, volume 6, pages 96–103, San Francisco, CA. Morgan Kaufmann Publishers.

Grimson, E. (1990). *Object Recognition by Computer.* MIT Press, Cambridge, MA.

Grimson, E. and Lozano-Perez, T. (1987). Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:468–482.

Hinton, G., Williams, C., and Revow, M. (1992). Adaptive elastic models for hand-printed character recognition. In Moody, J., Hanson, S., and Lippmann, R., editors, *Advances in Neural Information Processing Systems*, volume 4, pages 512–519, San Francisco, CA. Morgan Kaufmann Publishers.

Hummel, R. and Wolfson, H. (1988). Affine invariant matching. In *Proceedings of the Image Understanding Workshop*, pages 351–364, Cambridge, MA. DARPA.

Kosowsky, J. and Yuille, A. (1994). The invisible hand algorithm: Solving the assignment problem with statistical physics. *Neural Networks*, 7:477–490.

Lamdan, Y., Schwartz, J., and Wolfson, H. (1988). Object recognition by affine invariant matching. In *Conference on Computer Vision, Pattern Recognition*, pages 335–344. IEEE Press.

Leclerc, Y. G. (1989). Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3(1):73–102.

Lu, C. P. and Mjolsness, E. (1994). Two-dimensional object localization by coarse-to-fine correlation matching. In Cowan, J. D., Tesauro, G., and Alspector, J., editors, *Advances in Neural Information Processing Systems*, volume 6, pages 985–992, San Francisco, CA. Morgan Kaufmann Publishers.

Manmatha, R. (1994). A framework for recovering affine transforms using point, lines or image brightness. In *Conference on Computer Vision, Pattern Recognition*, pages 141–146. IEEE Press.

Ogawa, H. (1986). Labeled point pattern matching by Delaunay triangulation and maximal cliques. *Pattern Recognition*, 19:35–40.

Papadimitriou, C. and Steiglitz, K. (1982). *Combinatorial Optimization.* Prentice-Hall, Inc., Englewood Cliffs, NJ.

Peterson, C. and Soderberg, B. (1989). A new method for mapping optimization problems onto neural networks. *International Journal of Neural Systems*, 1(1):3–22.

Ranade, S. and Rosenfeld, A. (1980). Point pattern matching by relaxation. *Pattern Recognition*, 12:269–275.

Sclaroff, S. and Pentland, A. (1993). A modal framework for correspondence and description. In *Conference on Computer Vision, Pattern Recognition*, pages 308–313. IEEE Press.

Scott, G. and Longuet-Higgins, C. (1991). An algorithm for associating the features of two images. *Proc. Royal Society of London*, B244:21–26.

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

Sinkhorn, R. (1964). A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statist.*, 35:876–879.

Stockman, G. (1987). Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, 40:361–387.

Ton, J. and Jain, A. (1989). Registering Landsat images by point matching. *IEEE Transactions on Geoscience and Remote Sensing*, 27(5):642–651.

Umeyama, S. (1993). Parameterized point pattern matching and its application to recognition of object families. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15:136–144.

Vinod, V. and Ghose, S. (1993). Point matching using asymmetric neural networks. *Pattern Recognition*, 26:1207–1214.

Yuille, A. L. (1990). Generalized deformable models, statistical physics, and matching problems. *Neural Computation*, 2(1):1–24.