Optimal Affine-Invariant Point Matching
Mauro S. Costa, Robert M. Haralick and Linda G. Shapiro
Intelligent Systems Laboratory
Department of Electrical Engineering • FE-10
University of Washington – Seattle, WA 98115

Abstract
The affine-invariant matching scheme proposed by Hummel and Wolfsion [1] can be very efficient in a model-based matching system, not only in terms of the computational complexity involved, but also in terms of the simplicity of the method. This paper addresses the implementation of the affine-invariant point matching, applied to the problem of recognizing and determining the pose of sheet metal parts. We discuss errors that can occur with this method due to quantization, stability, symmetry, and noise problems. These errors make the original affine-invariant matching technique unsuitable for use on the factory floor. By beginning with an explicit noise model, which the Hummel and Wolfsion technique lacks, we then derive an optimal approach which overcomes these problems. We also evaluate the performance of the new algorithm under the influence of several distorting parameters.

1. Introduction
Affine-invariant matching is an appropriate technique for recognising flat or nearly flat objects in a 2D perspective projection image. This problem of object recognition is one of the most important tasks in computer vision. In an automatic manufacturing environment, it is necessary to recognize industrial parts as well as their positions and orientations in order for a robot to manipulate them. Most practical industrial robot vision systems are model-based systems in which well-defined, known models are matched against the image of a scene. The matching technique addressed in this paper is point matching.

The Hummel and Wolfsion affine-invariant matching technique makes use of local features to detect distinctive points. The distinctiveness of points is based on sharp convexities and deep concavities along the boundary of the objects. The distinctive points of a flat object can be modeled as a set \( M = \{ (x_m, y_m, \gamma) \}_{m=1}^M \) of model points all lying in the \( \gamma \) plane. Since the object is flat, an image of the object under perspective projection will have a corresponding set of observed points \( O = \{ (x_m, y_m) \}_{m=1}^M \) such that

\[
\begin{bmatrix}
x_m \\
y_m
\end{bmatrix} = \begin{bmatrix}
x_t \\
y_t
\end{bmatrix} + t,
\]

where \( R \) is a 2 × 2 rotation matrix \( t \) is a translation vector, and the pair \( (R, t) \) defines an affine transformation.

The affine transformation of a plane is uniquely defined by the transformation of three non-collinear points. Moreover, there is a unique map of any non-collinear triplet (here called a basis) in the plane to another non-collinear triplet; this mapping is defined by the affine transformation of the plane which contains the original triplet.

The most important observation is that for each non-collinear basis triplet, the coordinates of all other points in the plane, given in the coordinate system of the basis triplet, are affine invariant. If \( a, b \), and \( c \) are three non-collinear points in a plane, each represented by a \( 2 \times 1 \) vector, then any other point \( v \), with affine coordinates \( (\xi, \eta) \) with respect to basis \( < a, b, c > \), will still have coordinates \( (\xi, \eta) \) if the entire plane undergoes the affine transformation \( T \), assuming that the same triplet of transformed points \( < Ta, Tb, Tc > \) is chosen as the basis.

1.1 The Original Matching Algorithm
Given that the affine transformation is uniquely defined as the transformation of three non-collinear points in the plane, one can try to match each non-collinear triplet in the set of model interest points against non-collinear triplets in the set of scene interest points. The original algorithm [1] consists of two major steps: a pre-processing step and a recognition step.

In the off-line pre-processing step, for each ordered set of three non-collinear points selected from the set of model interest points, the affine coordinates of each of the remaining points with respect to these three basis points are computed. Each time a pair of affine transformed coordinates is computed, these values are quantized and used as an entry to a hash table, where the basis triplet and the model from which these coordinates came are recorded.

In the on-line recognition step, we are given a set of points that represent the projection of the object (or objects) in the image. Starting with any ordered non-collinear basis triplet of image points, the transformed coordinates of each of the remaining points are compared, just as in the pre-processing stage. Then each affine invariant coordinate votes for the closest (model, basis-triplet) triplets to it. Let \( (\xi(m; i, j, k), \eta(m; i, j, k)) \) be the affine-invariant coordinates of the \( m \)th model point with respect to a model basis \( < i, j, k > \), \( (\alpha(m; a, b, c), \beta(m; a, b, c)) \) be the affine-invariant coordinates of the \( n \)th image point with respect to an image basis \( < a, b, c > \), and \( q \) be the quantization function. Then, for each image basis \( < a, b, c > \), the Hummel and Wolfsion voting produces the count \( V(i, j, k; a, b, c) \) for basis model \( < i, j, k > \) defined by:

\[
V(i, j, k; a, b, c) = \# \{ n \mid q(\alpha(m; a, b, c), \beta(m; a, b, c)) = q(\xi(m; i, j, k), \eta(m; i, j, k)) \}.
\]

In operation, the quantized affine invariant coordinates of each image point with respect to image basis \( < a, b, c > \) accesses the hash table where it may pick up a pointer to a model basis having some model point whose affine coordinates hash to the same bucket, if there is such a one. The votes are then added into an accumulator associated with the model basis. When all the votes have been cast, the algorithm chooses to see if any model basis-triplet scored high enough. If no model basis-triplet achieves a high enough score, it means that the image basis-triplet selected in the set of image interest points does not correspond to any triplet in the set (or sets, in the case of more than one model) of model interest points. Another ordered basis-triplet in the image is then used, and the above procedure is repeated un-
1.2 Shortcomings of the Affine–Invariant Matching Technique

The affine-invariant matching technique is mathematically sound in the noiseless case. However, it has a number of shortcomings in practice. The problems we have discovered are as follows: 1) If the three non-collinear points selected as a basis are not numerically stable with respect to the other points, the coordinates of the transformed points are not reliable. 2) On real images, the coordinates of the detected interest points are noisy. This causes the wrong bin of the hash table to be accessed and produces unreliable results. 3) Partial object symmetries may cause bins representing incorrect transformations to have counts as high or nearly as high as the bin representing the correct transformation.

Problem 1 was solved with a simple test for determining the stability of a basis: if the area of the triangle defined by the three basis points is very small compared to the area of the object (in practice, the area of its convex hull), that basis is considered unstable and it is not used. A simple heuristic rule was used as an attempt to solve the quantization problems of the affine coordinates of the noisy image: for each interest point whose quantized affine coordinates are \((i,j)\), we only count votes for the tuples associated with bucket \((i,j)\), but also count partial votes for each one of its eight neighbors. This rule solved the accuracy and symmetry problems only partially. The details of the above solutions can be found in [2].

2. An Explicit Noise Model and Optimal Voting

We felt that the unsatisfactory performance of the technique was due to the lack of an explicit noise model; the original procedure did not model and therefore could not handle errors in the data. The procedure described below was developed on the premise that real data always contain errors which can be modeled, allowing the procedure to achieve satisfactory results.

Let \(<i,j,k>\) represent a model basis and \((\hat{\alpha}, \hat{\beta}), n = 1, ..., N-3\) represent the affine invariant coordinates of the observed image points with respect to a given image basis, which in this discussion we hold fixed. The question we answer here is what is the best way for the observations \((\hat{\alpha}, \hat{\beta}), n = 1, ..., N-3\) to vote for a model basis \(<i,j,k>\)? We suggest that for the votes to be meaningful, they should be related to

\[
P(<i,j,k> | (\hat{\alpha}, \hat{\beta}), n = 1, ..., N-3).
\]

Now, by the definition of conditional probability, and assuming that the prior probabilities for a model basis are equal, we have

\[
P(<i,j,k> | (\hat{\alpha}, \hat{\beta}), n = 1, ..., N-3) = \frac{P((\hat{\alpha}, \hat{\beta}), n = 1, ..., N-3 | <i,j,k>, \sigma, \xi, \nu, \delta)}{\sum_{<i',j',k'>} P((\hat{\alpha}, \hat{\beta}), n = 1, ..., N-3 | <i',j',k'>)}
\]

Concentrating on the key term \(P((\hat{\alpha}, \hat{\beta}), n = 1, ..., N-3 | <i,j,k>\), we assume that the observed affine invariant coordinates for those image points which arise from model points or which are unrelated to model points are conditionally independent of the given model basis \(<i,j,k>\). Hence,

\[
P((\hat{\alpha}, \hat{\beta}), n = 1, ..., N-3 | <i,j,k>) = \prod_{n=1}^{N-3} P((\hat{\alpha}, \hat{\beta}) | <i,j,k>).
\]

Since we desire to relate the observed affine invariant coordinates to a model point, if it can be so related, we let \(B_{<i,j,k>}\) be the set of model points excluding the basis point \(i, j,\) and \(k\), and \(m\) be a point in that set, that is, \(m \in B_{<i,j,k>}\). We then define the terms of the product by

\[
P((\hat{\alpha}, \hat{\beta}) | <i,j,k>) = \sum_{m \in B_{<i,j,k>}} P((\hat{\alpha}, \hat{\beta}) | m, <i,j,k>) P(m | <i,j,k>).
\]

We assume that the conditional probability of a model point \(m\) occurring in the image is independent of basis \(<i,j,k>\). Hence,

\[
P(<i,j,k> | (\hat{\alpha}, \hat{\beta}), n = 1, ..., N-3) = \prod_{n=1}^{N-3} \sum_{m \in B_{<i,j,k>}} P((\hat{\alpha}, \hat{\beta}) | m, <i,j,k>) P(m),
\]

where \(\Delta\) is a normalizing constant to make the probability sum to one when summed over all bases. In order to model the probability density \(P((\hat{\alpha}, \hat{\beta}) | m, <i,j,k>)\), two distinct cases must be taken into account: first, the case where the observed coordinates \((\hat{\alpha}, \hat{\beta})\) arise from a point which appears in the model; and second, the case where the observed coordinates \((\hat{\alpha}, \hat{\beta})\) arise from a point which does not appear in the model. In the second case, the image point giving rise to affine coordinates \((\hat{\alpha}, \hat{\beta})\) is an extraneous point. If we define a Bernoulli random variable, \(y\) given by

\[
y = \begin{cases} 1 & \text{if } (\hat{\alpha}, \hat{\beta}) \text{ comes from a point in the model} \\ 0 & \text{otherwise} \end{cases},
\]

the above probability density can be written as

\[
P((\hat{\alpha}, \hat{\beta}) | m, <i,j,k>) = P((\hat{\alpha}, \hat{\beta}), y = 1 | m, <i,j,k>) + P((\hat{\alpha}, \hat{\beta}), y = 0 | m, <i,j,k>).
\]

Taking into account that neither \(y\) nor the probability of observing affine coordinates \((\hat{\alpha}, \hat{\beta})\) which do not arise from the model do not depend on the model basis \(<i,j,k>\) or the model point \(m\), the two terms on the right-hand side of equation (2) can be rewritten as

\[
P((\hat{\alpha}, \hat{\beta}), y = 1 | m, <i,j,k>) = P((\hat{\alpha}, \hat{\beta}) | y = 1, m, <i,j,k>) P(y = 1).
\]

and

\[
P((\hat{\alpha}, \hat{\beta}), y = 0 | m, <i,j,k>) = P((\hat{\alpha}, \hat{\beta}) | y = 0) P(y = 0).
\]

Now, let \(P(y = 1) = q\). Consequently, \(P(y = 0) = 1 - q\). Also, let

\[
P((\hat{\alpha}, \hat{\beta}) | y = 1, m, <i,j,k>) = D_1
\]
and
\[ P\{\tilde{\alpha}_m, \tilde{\beta}_n \mid y = 0 \} = D_2. \]

Equation (2) can then be written as
\[ P\{\tilde{\alpha}_m, \tilde{\beta}_n \mid m, < i, j, k > \} = D_1 g + D_2 (1 - g). \]  

(5)

The quantity \( q \) is the probability that the observed affine coordinates \((\tilde{\alpha}_m, \tilde{\beta}_n)\) come from a point that appears in the model. Letting \( z \) be a variable denoting the number of model points that might appear in the image, \( q \) can be written as
\[ q = \sum_z P\{y = 1, z\} = \sum_z P\{y = 1 \mid z\} P\{z\}. \]

(6)

Let \( r \) be the probability that a model point appears in the image, \( M \) be the number of model points, and \( N \) be the number of image points. Then, in equation (6), \( P\{y = 1 \mid z\} \) is given by the following binomial distribution
\[ P\{y = 1 \mid z\} = \binom{M}{z} r^z (1 - r)^{M-z}, \]

and \( P\{z \mid < i, j, k >\} = \frac{z}{N} \). Rewriting equation (6) we obtain
\[ q = \sum_{z=0}^{M} \binom{M}{z} r^z (1 - r)^{M-z}. \]

But \( \sum_{z=0}^{M} \binom{M}{z} r^z (1 - r)^{M-z} \) is the expected value of a binomial distribution having parameters \( M \) and \( r \), and it is equal to \( Mr \). Therefore, \( q \) is finally given by \( q = \frac{M N}{N^2} \). To determine \( r \), we estimate the number \( L \) of model points which are not likely to appear in the image. Then \( r \) can be estimated by \( \frac{M}{N-L} \).

The two densities \( D_1 \) and \( D_2 \) in equation (5) can be modeled as follows. \( D_1 \) denotes the probability of observing the affine coordinates \((\tilde{\alpha}_n, \tilde{\beta}_n)\) which do not arise from the model. Therefore, we take it to be given by \( D_2 = C \), where \( C \) is a constant related to the area of the affine plane in which we expect values. Let \((\alpha_m, \beta_m)\) be the affine invariant coordinates of the \( m \)-th model point with respect to basis \(<i, j, k>\). In this case we assume normal errors:
\[ D_1 = \frac{1}{2\pi |\Sigma_m|^2} \exp\left\{ -\frac{1}{2} \left( \begin{array}{c} \tilde{\alpha}_n - \alpha_m \\ \tilde{\beta}_n - \beta_m \end{array} \right)^T \Sigma_m^{-1} \left( \begin{array}{c} \tilde{\alpha}_n - \alpha_m \\ \tilde{\beta}_n - \beta_m \end{array} \right) \right\}. \]

(7)

To determine the covariance \( \Sigma_m \) we note that the affine invariant coordinates \((\tilde{\alpha}, \tilde{\beta})\) of a point \((v_x, v_y)\) with respect to basis \(<a, b, c>\) are given by
\[ \tilde{\alpha} = \frac{(v_x - c_y)(v_y - c_x) - (v_x - c_y)(v_y - c_x)}{(a_x - c_x)(v_y - c_x) - (a_y - c_y)(v_x - c_y)} \]
\[ \tilde{\beta} = \frac{(v_x - c_y)(v_y - c_x) - (v_x - c_y)(v_y - c_x)}{(a_x - c_x)(v_y - c_y) - (a_y - c_y)(v_x - c_y)} \]

(8)

(9)

indicating that each of them is a function of eight variables. Thus, we can write
\[ \tilde{\alpha} = f(z_1, ..., z_8) \quad \text{and} \quad \tilde{\beta} = g(z_1, ..., z_8). \]

(10)

where each one of the eight random variables \(z_i\) can be expressed as \(z_i = \tilde{z}_i + \xi_i\), where \(\tilde{z}_i\) is the true unknown value of \(z_i\) and \(\xi_i\) is a random perturbation (noise) added to \(\tilde{z}_i\). Making the assumption that the \(\xi_i\) are independent and identically distributed with zero mean and standard deviation \(\sigma_i\) we write
\[ E[\xi_i] = 0 \quad \text{and} \quad E[\xi_i^2] = \sigma_i^2. \]

(11)

Therefore, equation (10) can be rewritten as
\[ \tilde{\alpha} = f(\tilde{z}_1 + \xi_1; \xi_1; i = 1, ..., 8) \quad \text{and} \quad \tilde{\beta} = g(\tilde{z}_1 + \xi_1; \xi_1; i = 1, ..., 8). \]

If we linearize these functions, by expanding them in a Taylor series and neglect second and higher order terms, we obtain
\[ \tilde{\alpha} = f(z_1, ..., z_8) + \sum_{i=1}^{8} \xi_i \frac{\partial}{\partial z_i} f(z_1, ..., z_8) \]
\[ \tilde{\beta} = g(z_1, ..., z_8) + \sum_{i=1}^{8} \xi_i \frac{\partial}{\partial z_i} g(z_1, ..., z_8). \]

(12)

(13)

In the above equations, note that \(f(\tilde{z}_1, ..., z_8) = \alpha\) and \(g(z_1, ..., z_8) = \beta\), since the values \(\tilde{z}_i\) are the coordinates of the points without noise (model). From equation (12), we can calculate the expected value of \(\tilde{\alpha}\) as follows
\[ E[\tilde{\alpha}] = E[f(\tilde{z}_1, ..., z_8)] + E[\sum_{i=1}^{8} \xi_i \frac{\partial}{\partial z_i} f(z_1, ..., z_8)] = \alpha \]

(14)

The variance of \(\tilde{\alpha}\) is given by
\[ \sigma^2_\alpha = E[(\tilde{\alpha} - E[\tilde{\alpha}])^2]. \]

(15)

From this point on, for simplicity, the functions \(f(\tilde{z}_1, ..., z_8)\) and \(g(z_1, ..., z_8)\) will be indicated in a vector notation by \(f(X)\) and \(g(X)\), respectively. Substituting equations (12) and (14) into equation (15) and using equation (11) the expression for \(\sigma^2_\alpha\) is found to be
\[ \sigma^2_\alpha = \sum_{i=1}^{8} \sum_{j=1}^{8} \frac{\partial}{\partial z_i} f(X) \frac{\partial}{\partial z_j} g(X) E[\xi_i \xi_j] \]

(16)

Assuming that the noise distribution for all points coordinates is the same, i.e., they have the same mean (zero) and same standard deviation \(\sigma_i\), and knowing that
\[ E[\xi_i \xi_j] = \begin{cases} \sigma_i^2 & ; i = j \\ 0 & ; i \neq j \end{cases} \]

(17)

since \(\xi_i\) are independent random variables, equation (16) becomes
\[ \sigma^2_\alpha = \sigma_i^2 \sum_{i=1}^{8} \left( \frac{\partial}{\partial z_i} f(X) \right)^2. \]

(18)

Following the above procedure for \(\tilde{\beta}\), it can be shown that the expected value and variance of \(\tilde{\beta}\) are given by
\[ E[\tilde{\beta}] = \beta \quad \text{and} \quad \sigma^2_\beta = \sigma_i^2 \sum_{i=1}^{8} \left( \frac{\partial}{\partial z_i} g(X) \right)^2. \]

(19)

The computation of the covariance of the two random variables \(\tilde{\alpha}\) and \(\tilde{\beta}\) is also similar and is given below. The covariance of \(\tilde{\alpha}\) and \(\tilde{\beta}\) is given by
\[ \sigma^2_{\alpha,\beta} = E[(\tilde{\alpha} - E[\tilde{\alpha}])(\tilde{\beta} - E[\tilde{\beta}])]. \]

(20)
Substituting equations (12), (13), (14) and (19) into equation (20) yields
\[ \sigma_{\alpha,\beta}^2 = \sum_{i=1}^{n} \frac{\partial f(X)}{\partial \alpha_i} \frac{\partial g(x)}{\partial \alpha_i}. \] (21)

The covariance matrix \( \Sigma \) associated with \( \alpha \) and \( \beta \) is given by
\[ \Sigma = \begin{pmatrix} \sigma_{\alpha}^2 & \sigma_{\alpha,\beta} \\ \sigma_{\alpha,\beta} & \sigma_{\beta}^2 \end{pmatrix} = \sigma_i^2 \begin{pmatrix} A & C \\ C & B \end{pmatrix}, \] (22)

The values \( A, B, C \) and \( D \) can be calculated by computing the partial derivatives of the functions defined in equations (8) and (9) [2].

To use the above optimal voting procedure, the voting accumulators must keep track of model point \( m \), image point \( n \), and basis triplet \( \langle i, j, k \rangle \). The actual vote \( V \) is a computed probability density:
\[ V(<i, j, k>, m, n) = P\{\langle \alpha_n, \beta_n \rangle \mid m, <i, j, k>\} \cdot P\{m\}. \]

We take \( P\{m\} = \frac{1}{M} \). Notice that in this formulation there is no required quantization of the affine coordinates or any explicit statement requiring that closeness of image affine coordinates with model affine coordinates be determined through a hashing function. However, to save computation, in our implementation, we retained the use of a hashing function to get us to an explicit set of linked lists of model bases, and affine-invariant coordinates with respect to a model basis of all model points lying close enough to the affine invariant coordinates of the image point. In this case, the computation for \( P\{\langle \alpha_n, \beta_n \rangle \mid m, <i, j, k>\} = P\{\langle \alpha_m, \beta_m \rangle \mid m, <i, j, k>\} \cdot \Sigma_m, <i, j, k> \). as given by equation (5), was only done for those bases and model points on those linked lists. The criteria used for selecting those linked lists is as follows. Lists which are considered to contain model bases for which \( P\{\langle \alpha_m, \beta_m \rangle \mid m, <i, j, k>\} \) is not significantly different from zero must be associated with buckets in the hash table that overlap with a given elliptical cross section of the Gaussian density of equation (7). This cross section is chosen to be the one for which the value of the density is equal to 90% of its peak value. To speed up the voting process, all the lists associated with the square neighborhood of buckets which encloses the selected ellipse are used for voting.

After the voting process determines that a given model basis has a high enough probability of matching a given image basis, a verification step has to take place. This step is necessary to ensure that there really exists a true correspondence between model points and image points, as established by the bases correspondence. The verification algorithm incorporates a statistical error model similar to the one used for the voting process. By defining such an error model, we avoid the use of arbitrary thresholds, such as the percentage of matching points between transformed model and image, and the results of the verification step are considerably more accurate and reliable.

3. Experiments

The main objective of the experiments performed is to characterize the performance of the technique under the influence of several parameters (noise level, number of interest points, number of missing and extraneous interest points, objects' symmetry, etc.) that may affect the outcome of the matching procedure. In order to achieve such characterization, five original models are used. Each of these models is distorted to generate four test models which present different combinations of the parameters number of interest points and symmetry. The observation sets are generated from the 20 test models, according to the following procedure. First, a random affine transformation is applied to the test model being used; then the points are distorted by adding noise to their \( x \) and \( y \) coordinates; finally some points are randomly removed and some points are randomly added to the set. The observation sets generated (6,480 total) correspond to all possible combinations of the following parameters: a) \( K \) - number of original models = 5; b) \( S \) - different noise levels (standard deviation) = 6 (0, 1, 3, 5, 10, 20); c) \( M \) - different number of test model points = 2 (10, 14); d) \( L \) - different number of missing points = 3 (0, 3, 5); e) \( E \) - different number of extraneous points = 3 (0, 5, 10); f) \( Z \) - symmetry = 2 (symmetric, non-symmetric); and g) \( R \) - number of replications of an experiment with a given combination of the above parameters = 6.

For each separate set of parameters \( S, L, E \), for each one of the 20 test models, and for each of the six \( R \)'s, the following is computed: 1) the probability \( P \) that the basis accepted by the algorithm is correct; 2) the percentage of bases \( n \) that failed to either threshold success (\( P \geq 0 \)) or failure (all stable bases yielded \( P < 0 \)); 3) the percentage of bases \( n \) tried after threshold success that did not verify before complete success or failure; 4) the percentage of bases tried \( n \) after threshold success and that achieved successful verification, but are not a correct solution; 5) the percentage of experiments that failed after trying all bases. The results, which exhibited by graphs of \( n_p, n_m, n_e \), and \( f \) as a function of \( S, L, \) and \( E \), averaged over all the combination of the other parameters, reveal that the technique performs quite well. The general trend is a decrease in performance (i.e., an increase in number of bases tried and failed) with the increase of either the noise, the number of missing points or the number of extra points (or a combination of them). The average percentage of bases that were attempted before a correct solution was found as a function of the standard deviation, missing points and extra points have been plotted, and they are approximately linear functions. The percentage of bases attempted before a correct solution is found is about 15% or less. The overall percentage of complete failure (cases for which no bases yielded \( P > 0 \)) was equal to 11.6%. It is important to state that in all cases for which a correct solution was found, the value of the parameter \( P \), which measures the probability of a match, was always equal to 1, reinforcing the strength of the technique.

4. References
