Optimal affine–invariant matching: performance characterization

Mauro S. Costa, Robert M. Haralick, and Linda G. Shapiro

Intelligent Systems Laboratory
Department of Electrical Engineering • FT-10
University of Washington, Seattle, WA

ABSTRACT

The Geometric Hashing scheme proposed by Lamdan and Wolfson [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. In a recent paper, Costa, Haralick and Shapiro [2], we discussed errors that can occur with this method due to quantization, stability, symmetry, and noise problems. These errors make the original geometric hashing technique unsuitable for use on the factory floor. Beginning with an explicit noise model, which the original Lamdan and Wolfson technique lacks, we derived an optimal approach that overcomes these problems. We showed that the results obtained with the new algorithm are clearly better than the results from the original method. This paper addresses the performance characterization of the geometric hashing technique, more specifically the affine–invariant point matching, applied to the problem of recognizing and determining the pose of sheet metal parts.

The experiments indicate that with a model having 10 to 14 points, with 2 points of the model undetected and 10 extraneous points detected, and with the model points perturbed by Gaussian noise of standard deviation 3 (0.58% of range), the average amount of computation required to obtain an answer is equivalent to trying 11% of the possible three-point bases. The misdetection rate, measured by the percentage of correct bases matches that fail to verify, is 0.9%. The percentage of incorrect bases that successfully produced a match that did verify (false alarm rate) is 13%. And, finally, 2% of the experiments failed to find a correct match and verify it. Results for experiments with real images are also presented.

1. Introduction

Affine–invariant matching is an appropriate technique for recognizing flat or nearly flat objects in a 2D perspective projection image. 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.

A brief summary of the affine invariant matching technique is described in Section 2. Section 3 discusses the experimental protocol utilized to characterize the performance of the technique. Results of the technique applied to real images are presented in Section 4.

2. The Affine–Invariant Matching Technique

The affine–invariant matching technique requires the 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.
2.1 Affine Transformation of Points in a Plane

It is well known that a necessary and sufficient condition to uniquely define a plane is a set of three non-collinear points in space. Consequently, the affine transformation of the plane is also 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 as a \( 2 \times 1 \) vector, then any other point \( v \), also represented by a \( 2 \times 1 \) vector, will have the same 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.

The mathematical representation of a generic point \( v \), in terms of its affine coordinates \( (\xi, \eta) \) and the basis triplet \( < a, b, c > \) which defines the plane, is given by the following equation:

\[
\begin{bmatrix}
\xi \\
\eta
\end{bmatrix} = \begin{pmatrix}
a & b - c \\
0 & 1
\end{pmatrix}^{-1} \begin{bmatrix}
v - c
\end{bmatrix}.
\]

Hence, given the affine invariant coordinates and the basis, the given point may be computed by

\[
v = \xi (a - c) + \eta (b - c) + c.
\]

Notice that if point \( v \) and its basis are transformed by \( T \), its new affine coordinates are

\[
\begin{bmatrix}
\xi \\
\eta
\end{bmatrix} = \begin{pmatrix}
T(a - c) & T(b - c)
\end{pmatrix}^{-1} T \begin{bmatrix}
v - c
\end{bmatrix}
\]

\[
= \begin{pmatrix}
T(a - c) & T(b - c)
\end{pmatrix}^{-1} T \begin{bmatrix}
v - c
\end{bmatrix}
\]

\[
= \begin{pmatrix}
a - c & b - c
\end{pmatrix}^{-1} \begin{bmatrix}
v - c
\end{bmatrix}
\]

which clearly shows that its transformed coordinates are affine-invariant.

2.2 The 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 non-collinear triplets in the set of model interest points against non-collinear triplets in the set of scene interest points. The algorithm consists of two major steps: a preprocessing step and a recognition step. The first step converts the model interest points into an affine invariant model representation. The second step, the matching proper, performs the same basic task, but now for the observed image points, and tries to match model against image using the affine representation. The result of the matching is a selection of models points, a selection of image points, and a one-to-one correspondence between the selected model points and the selected image points that indicates which of the selected image points arise from the affine transformation of the selected model points. The next two subsections describe these two steps in detail.
2.2.1 Preprocessing

In this off-line step, all possible combinations of stable, non-collinear triplets from the model are used as possible bases for the planar transformation. That is, for each ordered set of three stable, 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 points as the basis are computed using equation (1). 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, the model from which these coordinates came, and the affine coordinates themselves are recorded. This is done for as many models as needed. If new models have to be added to the database they can be processed independently, so that there is no need to recompute the hash table.

2.2.2 Recognition

In the on-line recognition step, we are given a set of interest points that represent the projection of the object (or objects) in the image. Starting with any ordered, stable, non-collinear basis triplet of image points, the transformed coordinates of each of the remaining image points are computed, just as in the preprocessing stage, and the voting takes place. In our optimal formulation, the actual vote is a computed probability density function [2], and in this formulation there is no required quantization of the affine coordinates nor 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 (associated with a particular bucket in the hash table), 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. When all the votes have been cast, that is, all probabilities have been computed, the algorithm checks 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 until a certain basis-triplet scores high enough (a match is declared to be found), or until all the possible combinations of image interest points have been tried as bases and no pair had a sufficient score (no match is found).

3. Experimental Protocol

The experimental protocol describes the experiments and analysis we perform to characterize the degree of correctness with which the affine-invariant point matching technique finds the correct match between an observation (image) basis and its corresponding model basis under various amounts of random noise distortions. Issues on user-defined and/or application-oriented performance criteria, based on the frequency of failure of the technique, are also considered.

A total of $K = 5$ different original models are used in the experiments which consist of thousands of replications under varying conditions. These models are sets of 2-D points extracted from CAD descriptions of objects after some low-level image processing and sets of points extracted directly from the image of a model object. Models 1, 2, 3, 4 and 5 and their interest points are depicted in Figure 1. Since the $x$ and $y$ coordinates of the interest points are generated from images, their values range from 0 to 511. A random affine transformation is applied to these models, and new models (called test models) are generated according to the procedure described in Section 3.1. Noisy observation sets are generated from these test models as explained in Section 3.2. These observation sets are not only noisy distortions of the point coordinates of the original models, but they also may contain extraneous points, and they may exclude some points from the model set as well.
Figure 1. Models (a) 1, (b) 2, (c) 3, (d) 4, and (e) 5 and their respective interest points.
3.1 Original Models and Test Models Generation

The sets of original model interest points are obtained from an interest points extractor. The models are selected so that among the $d$ model interest points, $g$ points ($g < \frac{d}{2}$) have mirror images with respect to some symmetry axis (in our experiments $g = 7$). The reason for using models that have this characteristic is that we can select suitable points that satisfy the symmetry/non-symmetry model parameter during the generation of the test models.

The test models are generated from the models according to the following procedure. For each original model, each one of its $d$ interest points is assigned an index which uniquely identifies it. Furthermore, each pair of symmetric points (points belonging to the symmetric subset of original model points) receive an identical label (different from the indices) that serves as a link between them. The points which do not belong to the symmetric subset of points receive no label.

To select $m$ points from among the original set of $d$ model points, we start by randomly selecting any point: we randomly generate an index from a discrete uniform distribution ranging from 0 to $d - 1$. If the test model being generated is to be symmetric, that point is only accepted if it has a label associated with it; if the test model is to be non-symmetric, that point is accepted regardless of its label. In the first case, the other model point which has the same label is added to the set being generated. We then randomly select another model point and repeat the process. If the test model is to be non-symmetric, we select at least one point with no label even if that means rejecting some points during the random generation of indices.

3.2 Observation Sets Generation

For each test model generated according to the procedure described above, an affine transformation $A$ presenting random degrees of rotation, scaling, skewing, and translation, is applied to its points coordinates. To ensure that the resulting transformed coordinates will lie in the interval [0;511] and that the area of the transformed model is within a known range, the transformation matrix is obtained as follows. The $x$ and $y$ coordinates of two points, $p_1$ and $p_2$, are randomly generated from a uniform distribution ranging from 0 to 511. The coordinates of a third point $p_3$ are similarly generated and if $p_1$, $p_2$, and $p_3$ are not collinear, $p_4$ is accepted, otherwise it is generated again until it satisfies the non-collinearity criterion. The coordinates of a fourth point, $p_4$, are computed from the coordinates of the three points already generated, such that $p_1$, $p_3$, $p_2$, and $p_4$, in this order, form a parallelogram. If either coordinate of $p_4$ lies outside the interval [0;511], points $p_3$ and $p_4$ are generated again until $p_4$ lies within the specified interval. If $p_4$ satisfies the criterion, and if the ratio of any two adjacent sides of the parallelogram obtained is within the interval [0.5;2], and if the angle between those two sides is greater than 60°, the four points generated are accepted, otherwise, the points generation procedure is repeated from the start until all criteria are satisfied. The transformation between the accepted parallelogram's corners and the corners of the rectangle which encloses the model (defined by the maximum and minimum values of the $x$ and $y$ coordinates of the model points), is determined to be the desired affine transformation. By using the above procedure, the relative area range of the transformed model is always between 0.5 and 2, and the transformed model points are geometrically arranged so that, most likely, they will not give rise to unstable bases [2].

After the model points are transformed, they are distorted by adding noise to their coordinates. The noise is generated by translating each transformed point by an independent translation vector $\vec{b}$, which is randomly generated from a Gaussian distribution with zero mean and standard deviation $s$. The standard deviation $s$ assumes one of the 6 possible values ranging from 0 to 20, as given in Section 3.3.2. If $\vec{C}$ is a vector denoting the coordinates of an interest point in the model and $\vec{O}$ is a vector denoting the corresponding point in the observation of that model, then

$$\vec{O} = A \vec{C} + \vec{b} + \vec{B}$$
where \( \bar{C} = \begin{pmatrix} c_x \\ c_y \end{pmatrix}, \bar{O} = \begin{pmatrix} o_x \\ o_y \end{pmatrix}, \bar{B} = \begin{pmatrix} b_x \\ b_y \end{pmatrix}, A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \) and \( \bar{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}. \)

To incorporate characteristics such as missing points and extra points in the noisy observation sets being generated, the following technique is used. The number of points in the observation set, \( n \), is given by

\[
n = m - l + e,
\]

where \( m \) is the number of test model points, \( l \) is the number of missing model points and \( e \) is the number of extraneous observation points.

Both \( l \) and \( e \) are integers with values as given in Section 3.3.2. A repeatedly generated discrete random variable uniformly distributed between 0 and \( m - 1 \) is used to obtain the \( l \) unique indices of the points that are to be missing in the observation set being constructed. To obtain the \( e \) extraneous points, we randomly generate from a discrete uniform distribution points whose coordinates range from 0 to 511. This yields extraneous points lying anywhere in the range of the standard image size.

### 3.3 The Experiments

For each observation set being matched against the respective test model, the conditional probability, denoted by \( p \), that a basis randomly selected from among the stable bases in the observation set corresponds to a given test model basis is computed [2]. If this probability is less than a given threshold \( t \), another basis is randomly selected and the process is repeated. If the probability is greater than or equal to the threshold, a possible match is considered to be found and a verification step, as described in [3], is carried out. If the verification is successful, the computed probability is used to characterize the performance of the method; if not, another basis is selected and the process repeated. This is done until a match is found or until all the stable image bases have been attempted.

In order to completely characterize the performance of the above algorithm, we are interested in the frequency with which the meaningful computational outcomes occur in the experiments. These outcomes are:

1) A basis with \( p < t \),
2) A basis with \( p \geq t \), unsuccessful verification and correct match,
3) A basis with \( p \geq t \), successful verification and incorrect match,
4) A basis with \( p \geq t \), successful verification and correct match.

For each experiment, image bases will be tried in a random fashion to establish the frequencies of these outcomes until outcome 5 is achieved.

The parameters involved in this investigation are listed below, with the respective individual values they can assume:

- \( K \) - number of original models = 5,
- \( S \) - different standard deviations of the noise = 6: 0, 1, 3, 5,
- \( R \) - different number of replications of a given experiment = 10,
- \( M \) - different number of test model points = 2: 10 and 14,
- \( L \) - different number of missing points = 3: 0, 2 and 4,
- \( E \) - different number of extraneous points = 3: 0, 10 and 20,
- \( Z \) - symmetry = 2: symmetric and non-symmetric,

where the quantity \( K \times M \times Z \) gives the number of test model variations. Therefore, a total of \( 5 \times 4 \times 6 \times 2 \times 3 \times 3 \times 2 \times 7 \times 200 \) experiments were performed.
3.4 Results

A success threshold $t$ of 0.85 was used in all the experiments. For each separate set of parameters $S$, $L$ and $E$, for each of the twenty test models, and for each of the ten replications of an experiment, we have computed the following:

1) the probability $P$ that the basis finally accepted by the algorithm is correct;
2) the percentage of bases tried before either threshold success ($P \geq t$) or failure (all stable bases yielded $P < t$);
3) the percentage of bases tried after threshold success and that did not achieve successful verification, but are a correct solution (misdetection rate);
4) the percentage of bases tried after threshold success and that achieved successful verification, but are not a correct solution (false alarm rate);
5) the percentage of experiments that ended in complete failure after trying all bases.

The results of the experiments are summarized in the form of graphs in Figures 2, 3, 4, and 5. The horizontal axis of each graph represents the standard deviation of the noise. The graphs of Figure 2 illustrate 2) above, the percentage of bases tried by the algorithm before either a successful basis was found or all bases led to failure. Figure 2(a) illustrates the relationship for zero extra points, Figure 2(b) for ten extra points, and Figure 2(c) for twenty extra points. The three different values for missing points (zero, two, and four) are illustrated by the three different curves in each of 2(a), 2(b), and 2(c). Each data point in every curve is the average of the results of 200 experiments corresponding to the 20 different test models and the 10 different replications of each experiment.

Figures 3, 4 and 5 illustrate items 3), 4), and 5) above, with the graphs following the same conventions as those in Figure 2. For example, in (b) of Figures 2, 3, 4, and 5, for a model with 10 or 14 points, having 2 model points not showing on the image and 10 extra model points showing up, with the model points showing up being perturbed by noise of standard deviation 3, about 11% of the total bases will be tried and will not produce a match, before a solution is found; among the bases that did produce a match and were indeed correct ones, 0.9% will fail to verify (misdetection rate); 13% of the incorrect bases that did produce a match will achieve successful verification (false alarm rate); and, 2% of the experiments with the above parameters will completely fail.

It is important to state that the cases that yielded a high percentage of incorrect verifications are due to the fact that the combination high noise and large number of extra points allows transformations which will align part of model well with the corresponding image points, but which will also align part of the model to an incorrect set of image points. The skewing portion of such transformations is greatly responsible for the incorrect alignment, as depicted in Figure 6. An example of completely random match is illustrated in Figure 7. In both examples, the matched sets successfully passed the chi squared verification test [3]. The effect of random matches is somewhat intrinsic to hashing techniques as it has been pointed out by Grimson and Huttenlocher in [4]. However, his analysis was done for the original Lamdan-Wolfson voting technique, and, therefore, it does not directly apply to our method. In [5], Lamdan and Wolfson claim that the probability of a random match is very low, but they did not take into consideration the cases of partially correct/incorrect alignments. It turns out that the probability of such cases is much higher than expected, as our results show.

It can be seen in the figures above that the overall behavior of the technique is as expected. 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). 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 voting technique.
Figure 2. Percentage of bases tried before success or failure as a function of the standard deviation of the noise.

Figure 3. Percentage of correct matches that failed to verify as a function of the standard deviation of the noise.

Figure 4. Percentage of verifications that were incorrect as a function of the standard deviation of the noise.
4. Experiments With Real Images

To test the performance of the technique on real data, three models were used: two pliers and a hammer. The technique was applied to the images shown in Figure 9. Figures 8 (a), (b), and (c) also show the interest points corresponding to the models of the objects. Since there were no CAD models of the objects used, the points were manually selected in the following fashion. The objects were placed on a piece of graph paper and the points determined by the human experimenter to be of high curvature (convexities/concavities) were plotted and recorded as model interest points. Figure 9 shows the boundaries and interest points of the test images that were detected by the interest operator. Each image was matched against a database composed of the three models, and the results, including the execution times in a Sun4/110 computer, are shown in Table 1. No effort was made to speed up the code.

Table 1. Results for Set of Real Images.

<table>
<thead>
<tr>
<th>Image</th>
<th>Model matched</th>
<th>Points matched (I,M)</th>
<th># Bases tried</th>
<th>Exec. time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><em>Plier1</em></td>
<td>(1,1) (9,4) (5,7)</td>
<td>6</td>
<td>0.978 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3,8) (2,11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><em>Plier2</em></td>
<td>(3,1) (1,2) (8,4)</td>
<td>4</td>
<td>0.700 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(7,5) (5,7) (4,9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><em>Hammer</em></td>
<td>(7,2) (6,3) (5,4)</td>
<td>3</td>
<td>0.731 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(9,9) (10,10) (8,11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><em>Plier1</em></td>
<td>(7,1) (19,4) (16,5)</td>
<td>12</td>
<td>3.178 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(15,9) (11,11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><em>Plier2</em></td>
<td>(21,1) (9,2) (5,6)</td>
<td>28</td>
<td>6.508 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3,7) (4,8)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 6. Shows a non-intuitive, but common kind of occurrence: an example of a partially random match. (a) test model generated from original model 5; (b) observation set generated: standard deviation of the noise=2, extra points=10, missing points=0; (c) correct instance of the model in the image (model points are indicated by • and non-model points are indicated by a ×); (d) random match found (incorrect points matched to the model are indicated by a • and points not matched are indicated by ×).
It can be seen that all matches found are correct. It is important to note that the models of Figure 9 had errors associated with the manual selection of interest points, but the technique was still able to recognize them in the images presented, even with no special adjustment of the input parameters, which were kept constant at the following values: probability of a success threshold = 0.85; estimated standard deviation = 2.0; estimated percentage of missing points = 20%; estimated probability that a point is extraneous = 0.1.

5. Conclusions

We have developed a procedure that starts with a CAD model of a flat or nearly flat object, constructs a vision model in the form of a hash table for affine-invariant matching, and is able to rapidly detect the pose of an instance of that object. We improved the original affine-invariant matching procedure by several additions. An explicit noise model was assumed and an optimal voting procedure was derived. A robust verification algorithm that models the noise in the data was also developed. Under the explicit noise model, the performance improved significantly. The experiments with synthetic data have been carefully designed and all parameters and conditions involved have been explicitly indicated, such that the experiments can be replicated by any interested researcher. The results of the experiments with synthetic data show that the frequency of failure of the technique was within acceptable ranges for the values of the affecting parameters that are expected when dealing with real images. The experiments with real image data prove that the technique works well, and that the problems with the original affine invariant matching technique have been effectively solved.

6. References


Figure 7. Shows a non-intuitive, but somewhat common kind of occurrence: an example of totally random match. (a) test model generated from original model 2; (b) observation set generated: standard deviation of the noise=3, extra points=10, missing points=0; (c) correct instance of the model in the image (model points are indicated by \( \bullet \) and non-model points are indicated by \( \times \)); (d) random match found (incorrect points matched to the model are indicated by \( \bullet \) and points not matched are indicated by \( \times \)).
Figure 8. Test images (the white dots in (a), (b) and (c) correspond to the models' interest points).
Figure 9. Interest points found by the interest operator.