# A FEATURE-BASED SCENE MATCHER

Donald B. Gennery

Robotics and Teleoperator Group
Jet Propulsion Laboratory
Pasadena, California 91103

## ABSTRACT

A method is presented for matching two scene descriptions, each of which consists of a set of measured feature vectors with estimated uncertainties. The two scenes differ by a transformation that depends on a few unknown parameters. The method performs a search by sequentially matching features of one scene to those of the other scene, solving for the transformation parameters by means of a generalized least-squares adjustment, computing the probabilities of these matches by means of Bayes* theorem, and using these probabilities to prune the search. An example is given using scene descriptions of the Martian surface in which the features are rocks approximated by ellipsoids.

## INTRODUCTION

A common problem in vision is that of matching. This may involve ma telling one two-dimensional image to another, matching a three-dimensional object to its two-dimensional projection, or matching two three-dimensional descriptions of the same scene. For want of a better term, the entities to be matched will be called "scenes". In general, the two scones may be related in various complicated ways. For example, there may be "rubber sheet" distortion [l], there may be independent changes in different parts of the scene as in the stereo matching problem [2], or there may be arbitrary changes in the different portions of a complicated object as in matching to a generic model [3]. However, in this paper the problem is restricted to the case in which the relationship between the scenes to be matched can be expressed by a few parameters, each of which has a global effect. Restricting the problem in this way allows the use of robust, efficient techniques that would not be applicable in the more general case.

It Is further assumed here that the matching will be done by using local features that have been detected in each scene, that the a priori correspondence between features in the two scenes is unknown, that each feature is described by a vector of measured feature parameter values, and that for each feature the probability distribution of its measured feature vector values is known. Although in principle any probability density function of errors in the feature vector could be used, the following treatment is restricted to the special case in which the error distribution is determined by its covariance matrix. The possibility exists that a feature detected in one scene will not be detected in the other, and this possibility is covered by estimates of probability of detection based on experience.

For example, consider the matching of two aerial photographs which differ by an unknown translation, rotation, and scale factor. In this case four parameters determine the transformation. The features might be local landmarks whose two-dimensional position and orientation have been measured, in which case each feature vector would have three components. Another example might be recognizing in a two-dimensional image the projection of a known three-dimensional object. Six parameters determine the position and orientation of the object. The features might be markings, corners, or edges that are known In the object model but must be found in the image.

For a third example, consider an object finder [2, 4] which produces a description of a three-dimensional scene in terms of objects represented by ellipsoids. Thus each feature vector consists of the nine parameters that define the ellipsoid. Suppose that the same area has been viewed from two different locations by a roving vehicle. There will be a partially unknown translation and perhaps rotation (requiring six parameters in general) between tne coordinate systems cf the resulting two scene descriptions, caused by errors in the navigation system of the vehicle. It is desired to match these two descriptions in order to correct the navigation data. (This special case is discussed further elsewhere [2], and an example of its use is presented later in this paper.)

By using the probability distributions of the feature vectors, it is possible to find the optimum matching of the scene in an efficient way that does not consider all possible matches of individual features and yet does not overlook matches

---

*The work described in this paper was started at Stanford University and was completed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

that have an appreciable chance of turning out to be correct. In this regard the method to be presented can be considered to be an improvement over the method of Price [5] and the suggestion of Milgram and Bjorklund [6]. It also has some overlap with the method of Fischler and Elschlager[1], lying somewhere between their dynamic programming method and their linear embedding algorithm. (In their terminology, a "spring" would connect each feature to some constant point in the scene for the problem stated above. This is not the kind of situation for which their method was designed to work well.)

It is assumed below that the matching of features is one-to-one (with perhaps some features left unmatched) and that all features are of the same type (so that a feature in one scene can be matched to any feature in the other scene). However, the mathematics can be generalized to the case where several features in one scene can be matched to the same feature in the other scene and to the case where there are different types of features that cannot be matched to each other (with perhaps a different size of feature vector for each type).

## NOTATION

Matrix notation will be used. (Hohn [7] provides a good text on matrix algebra.) For any matrix A, its transpose is denoted by $A^T$, its inverse by $A^{-1}$, and its determinant by det(A). Vectors (of any size) are represented by column matrices. The covariance matrix of any vector X is defined in the usual way to be

$£[(X-£x)\ (X-6*X)^T]$, where £ is the mathematical expectation operator. A partial derivative matrix of a vector Y with respect to a vector X is defined so that the rows correspond to elements of Y and the columns correspond to elements of X.

The two given scene descriptions are called Scene A (composed of features A1, A2, etc.) and Scene B (composed of features B1, B2, etc.). For each feature of each scene the following are given, with subscripts to denote particular features:

X = measured feature vector.

H = covariance matrix of X.

b • probability that the feature will be present in a matching scene.

r = a priori probability density of feature vector for typical scenes, evaluated at X but not otherwise using measured data.

(It is necessary that the dimensionality of Scene B be at least as great as that of Scene A, so that any feature in Scene B can be uniquely projected into Scene A if the transformation is known.)

The following global quantities are given:

$p_0$ - a priori probability that the two scenes match (that is, represent the same physical scene).

$G_0$ - a priori estimate of vector of parameters used in the transformation from Scene B to Scene A.

$\Gamma_0$ = covariance matrix of $G_0$.

It is desired to compute the following:

p = probability that the scenes actually match.

G = vector of transformation parameters (if the scenes match).

$\Gamma$ = covariance matrix of G.

The following definitions will be needed:

$F(G,X_{Bj})$ = feature $B_j$ projected from Scene B into Scene A, where the multidimensional function F is derivable from the known transformation between scenes and the definition of the feature vector.

$P_j$ = partial derivative matrix of $F(G,X_{Bj})$ with respect to G.

$R_j$ = partial derivative matrix of $F(G,X_{Bj})$ with respect to $X_{Bj}$.

(Ai,Bj) = pair denoting match of feature Ai to feature Bj.

M = set of (Ai,Bj) pairs for a possible matching of the two scenes (empty if no pairs are matched).

$M_A$ = set of Ai's contained in the members of M.

$M_B$ = set of Bj's contained in the members of M.

m = number of (Ai,Bj) pairs in M.

a = number of parameters in a Scene-A feature vector (size of $X_{Ai}$).

g = number of transformation parameters (size of G).

The symbol $\varepsilon$ denotes set membership.

Some important intermediate quantities are defined as follows:

$\pi_M$ = a priori probability that M would occur, given that the scenes match but ignoring the measured feature vectors.

$\rho_M$ = joint probability density of the feature vectors evaluated at the observed values, given that M is the correct matching.

$\rho_0$ = a priori joint probability density of the the feature vectors.

$P_M$ = a posteriori probability of the scenes matching and M being the correct set of matches, given the observed feature vectors.

Other intermediate quantities are defined as needed.

## OPTIMUM MATCH

The method to be used involves producing tentative matches of the features in one scene to those in the other and for each match M computing the a priori probability $\pi_M$, computing the transformation vector GM, and comparing the residuals of the fit to the given feature vector distributions to evaluate the probability density $p_M$. Then

Bayes' theorem is used to compute the a posteriori probability $p_M$ The M for which $p_M$ is near unity, if any, can be assumed to be the correct match, and the corresponding $G_M$ can be used for G.

Thus the a posteriori probability of M is computed by means of Bayes' theorem as follows:

$$p_M = \frac{p_o \pi_M \rho_M}{(1-p_o)\rho_o + p_o \sum \pi_M \rho_M} \tag{1}$$

since $p_{0IIM}$ is the a priori probability of the scene matching according to M, $1-p_0$ is the probability of the scenes not matching, and $p_M$ and $p_0$ respectively are the probability densities that the observed results (feature vectors) would follow from these events. (It would appear that all combinations of feature matches would have to be used in this computation, but a way of avoiding this will be described in the next section.) It remains to describe the computation of $\__{M'}$, $p_M$ and $p_o$.

The TTM quantities needed in (1) can be found by the following reasoning. The probability that feature Ai will be matched to some feature in Scene B is $b_{AI}$, the probability that it will be unmatched is $1-b_{Ai}$, and similarly for $b_{Bj}$ and $1-b_{Bj}$ for feature number j in Scene B. Thus the probability of a particular subset of m features from Scene A and m features from Scene B being matched, and no others, is the product of these terms over all features, chosen according to whether each feature is matched or not. However, there are m! ways of matching a set of m features to another set of m features (since the matching is assumed to be one-to-one). Therefore, this product is divided by m! to obtain the a priori probability of an individual matching combination. Thus

$$\pi_M = \frac{1}{m!} \prod_{Ai \in M_A} b_{Ai} \prod_{Ai \notin M_A} (1-b_{Aj}) \prod_{Bj \in M_B} b_{Bj} \prod_{Bj \notin M_B} (1-b_{Bj}) \tag{2}$$

The solution for the G parameters for a particular matching is computed as follows. An approximation to G, denoted by G', is used to compute the partial derivative matrices P and R of the transformation $F(G',X_B)$ as defined in the Notation section. (Note that P and R are implicit functions of G' and $X_B$, in general.) Then for this approximation a discrepancy vector U and its covariance matrix T for each matching of feature Ai with feature Bj is computed as follows:

$$U_{ij} = X_{Ai} - F(G',X_{Bj}) \tag{3}$$

$$T_{ij} = \Xi_{Ai} + R_j \Xi_{Bj} R_j^T \tag{4}$$

The solution for the parameters can be obtained by a generalized least-squares (minimum-variance) adjustment according to the general method described in [8] or in Appendix A of [2], which produces the maximum-likelihood solution if the errors have the normal (Gaussian)distribution. (This is only approximately correct here because of the fact that the parameters being adjusted affect the weights through $R_j$ in (4).) Thus,

$$H_M = \Gamma_o^{-1} + \sum_{(Ai,Bj) \in M} P_j^T T_{ij}^{-1} P_j \tag{5}$$

$$C_M = \Gamma_o^{-1}(G_o - G') + \sum_{(Ai,Bj) \in M} P_j^T T_{ij}^{-1} U_{ij} \tag{6}$$

$$D_M = H_M^{-1} C_M \tag{7}$$

$$G_M = G' + D_M \tag{8}$$

$$\Gamma_M = H_M^{-1} \tag{9}$$

If F is nonlinear, the above process may need to be iterated. (G' would be $G_0$ on the first iteration but later would be $G_M$ from the previous iteration.)

The above solution can be used to obtain $P_M$ for use in (1), provided that the distribution of the measurement errors is known. First,the residuals are

$$V_{ij} = U_{ij} - P_j D_M \tag{10}$$

If the errors are normally distributed, the multivariate normal distribution produces the following probability density $\rho_{ij}$:

$$\left.\begin{aligned} \rho_{ij} &= \frac{1}{(2\pi)^{a/2}\sqrt{\det(T_{ij})}} \exp\left(-\frac{1}{2} V_{ij}^T T_{ij}^{-1} V_{ij}\right), \\ &\qquad \text{if } (Ai,Bj) \in M \\ \rho_{io} &= r_{Ai}, \text{ if } Ai \notin M_A \end{aligned}\right\} \tag{11}$$

where $P_{10}$ is for the case in which feature Ai is unmatched. It includes the a priori distribution of Scene A features only, to be consistent with the discrepancies being computed in Scene A coordinates according to (3). (Strictly speaking, $p_{ij}$ for matched features should Include the effects of the a priori distribution also. However, since the a priori distribution is usually much wider than that given by T, this usually would have little effect.) Then, the probability density for the complete match is the product of all of the $p_{ij}$ values in this combination times the probability density function of the a priori parameter values, as follows:

$$\rho_M = \frac{1}{(2\pi)^{g/2}\sqrt{\det(\Gamma_o)}} \cdot \exp\left[-\frac{1}{2}(G_o - G_M)^T \Gamma_o^{-1} (G_o - G_M)\right] \prod_i \rho_{ij} \tag{12}$$

where the product is over all objects in Scene A.

An alternative way of computing $\rho_M$ is possible that may be more efficient than the above. Substituting (11) into (12) and combining the exponential functions produces an equation with the summation $\sum V_{ij}^T T_{ij}^{-1} V_{ij}$ in the exponent, where an extra term in the summation represents the a priori parameter values by letting $V_0 = G_0 - G_M$ and $T_0 = \Gamma_0$. This summation is the quadratic form which was minimized in the adjustment, and it can be shown [9] that this is equal to $\sum U_{ij}^T T_{ij}^{-1} U_{ij} - D_M^T C_M$, where as before the a priori parameter values are included in the summation by letting $U_0 = C_0 - C'$. Therefore, (12) is equivalent to the following when the errors have the normal distribution:

$$\rho_M = \frac{\exp\left\{-\frac{1}{2}\left[(C_o - G')^T \Gamma_o^{-1} (C_o - G') + \sum_{(Ai,Bj)\epsilon M} U_{ij}^T T_{ij}^{-1} U_{ij} - D_M^T C_M\right]\right\}}{\sqrt{(2\pi)^B \det(\Gamma_o) \prod_{(Ai,Bj)\epsilon M} (2\pi)^B \det(T_{ij})}} \prod_{Ai \notin M_A} r_{Ai} \tag{13}$$

In order to be consistent with the above, the a priori probability density of the feature vectors is computed using Scene A features only, as follows:

$$\Pi r_{Ai} \tag{1A}$$

(The $r_B$ values could be included here and in both parts of (11), but they would cancel out in (1).)

For each match combination, $\Pi_M$ from (2), $p_M$ from (12) or (13), and $p_0$ from (1A) are used in (1) to compute $p_M$. The $G_M$ and $T_M$ from (8) and (9) with the greatest probability $p_M$ can be used for the transformation parameters G and their covariance matrix T More accurately, if it is desired to include all matching possibilities and not just the most probable in the solution, it can be computed as follows:

$$\left.\begin{aligned} p &= \sum_M p_M \\ C &= \frac{1}{p} \sum_M p_M G_M \\ \Gamma &= \frac{1}{p} \sum_M p_M \Gamma_M + \frac{1}{p} \sum_M p_M (G_M - G)^2 \end{aligned}\right\} \tag{15}$$

## SEARCH PROCEDURE

With a large number of features, it would be impractical to use all combinations in (1) and (15X However, because of the exponential function in (11) or (13), most of the $p_M$ values (and hence $P_M$ values) will be negligibly small, and these terms can be ignored. The problem is to determine which combinations will produce significant magnitude in $p_M$ without having to compute them all.

The approach used is to select the features in Scene A one at a time and tentatively to match these to unmatched features in Scene B. The a posteriori probability of each of these partial M combinations is computed, and those with negligibly small probability are not pursued further. In this way a search tree is built up, branching out as different features in Scene B (including no feature) are matched to the current feature in Scene A at each level. The features in Scene A can be ordered according to a criterion based on their feature vectors so that those most likely to be unambiguously matched are matched first. Some kind of best-first search could be used, but in order to take advantage of Bayes' theorem in computing the probabilities for pruning the search, a breadth-first search is used instead.

Because, when a tentative match is made, it is not known which features in Scene B will remain unmatched, there is no obvious way to use all of the information in (2). Instead, $\Pi_M$ is computed by the following recursive method for the purposes of the search, where i now denotes a sequential numbering of Scene A features in the order in which they are selected for matching:

$$\left.\begin{aligned} &\pi_o = 1, \quad B_n - \underset{j}{5} >_{Bj} \\ &\pi_i = \pi_{i-1}\, b_{Ai} \frac{b_{Bj}}{\beta_{i-1}}, \quad \beta_i = \beta_{i-1} - b_{Bj}, \\ &\qquad\qquad \text{if } (Ai,Bj) \,\epsilon\, M \\ &\pi_i = \pi_{i-1}\,(1-b_{Ai}), \quad \beta_i = \beta_{i-1}, \text{ if } Ai \notin M_A \end{aligned}\right\} \tag{16}$$

The $\pi_i$ values from (16) are used for $\Pi_M$ in (1) for this particular node at the ith level of the search tree, with the other quantities in (1) being computed as described in the previous section, for this partial match. The resulting $p_M$ probabilities are used to prune the search.

The criterion used for pruning ideally should not use a constant probability threshold, but should take into account the fact that, if a large number of nodes have small probability each but sum to a large probability, there is a good chance that one of them will turn out to be part of the correct solution. An appropriate method would be to sort by probability all of the nodes at a given level in the search and to reject all of the nodes with smallest probabilities that sum to less than some threshold. A simpler (and more tolerant) method is to reject any node whose probability times the number $o_i$ nodes at this level is less than the threshold.

As the bottom level of the search tree Is reached, not all of the available information will have been used in computing ΠM by the above method. Thus $\Pi_M$ from (16) at the bottom level will act as an upper limit to the value that would have been obtained from (2). This effect causes less pruning to occur than the optimum computation would produce, but it should not result in the rejection of good solutions.

At each level of the search, the complete parameter computation according to (3) through (8) can be computed, including the iterations. However, the result from the parent node in the previous level of the search tree can be used as the initial approximation at this node, so that fewer iterations (perhaps only one) would be needed at each level.

Also, under some conditions the variation in the elements of $P_j$ with respect to G can be ignored. (This is likely to occur if $G_0$ is accurate.) In such a case the summations involving $P_j$.. In (5) and (6) can be computed recursively by adding the terms for a new $(A_i,B_j)$ combination to the total accumulated at the parent node, thus saving time (or a Kalman recursive estimation technique, also called sequential adjustment [8],could be used with slight additional time savings). In such a case, time can be saved also in the computation of PM if (13) is used, by recursively accumulating the sum of $U^T \; T^{-1} U$ and the product of $(2\pi)^a$ det $(T_{ij})$ as new combinations are added. C' in (13) would then be constant (perhaps equal to $G_0$). If $G_0$ is not accurate, it may be necessary to postpone the use of the recursive method until enough features have been matched to determine G well. (An alternate method of recursively computing $p_M$ is described in [2].)

In order to save time in the search phase, it may be desirable to use less than the full information that is available. For example, the feature vector can be collapsed to a vector with fewer components that capture most of the Information useful in discrimination and adjustment. This would result in slightly less effective pruning, but the time saved in computing each node may more than make up for this. The approximations of this type that can be made depend upon the application. Some are discussed for one case in [2].

In any case, the solution at the bottom level can be used as an initial approximation for one or more iterations with the full computation described in the previous section, using those combinations that have not been pruned because of small probability.

## EXAMPLE

As mentioned in the Introduction, one possible application for the method described in this paper is the matching of scene descriptions produced by the object finder described in [2] and [A], Such an example will now be presented.

A stereo pair of pictures (each about 10° by 10°) taken by the Viking Lander 1 on the surface of Mars was used as the Input to processing [2] consisting of stereo depth mapping, a ground finder, and the object finder. Figures 1 and 2 show the results when height thresholds of 5 centimeters and 3 centimeters, respectively, were used in the object finder. These figures show the ellipsoidal objects (approximating the rocks) projected into the left picture. (The large rock in the center is about 3.5 meters from the cameras and is about 0.3 meters long.) Each ellipsoid represents a feature to be used in the matching process. Arbitrarily assigned feature numbers are shown in the figures. It can be seen that more objects were detected in Figure 2 than in Figure 1 and that for those objects detected in both cases the ellipsoids are slightly different. Thus two different descriptions of the same scene are available for illustrating the operation of the matcher. (The poor fit to the upper right rock was caused by the fact that the background behind it was outside of the right picture, as explained in [2]. The white blob is an artifact of the picture-taking process.)

Although the object finder produces nine parameters to define each ellipsoid, only three parameters were used here in the matching for simplicity, consisting of the two components of horizontal position and the size. These three parameters formed the feature vector X, and the corresponding three-by-three covariance matrix = was derived from the nine-by-nine covariance matrix produced by the object finder. As a further simplification, the covarlances of size with position were ignored. The vector G of parameters to be adjusted consisted of the two components of horizontal translation, rotation in the horizontal plane, and scale factor. Translation was completely free to be adjusted (in effect the a priori values had infinite standard deviation), the a priori value for rotation was zero with a standard deviation of 1°, and the a priori value of scale lactor was unity with a standard deviation of 0.01. The b values, representing the probability of detection of a feature, were made a function the ellipsoid size, varying from zero for zero size up to 0.9 for a very large object. $p_0$ was set to 0.5. This information was used in a version of the matching algorithm implemented for this application and described in detail in [2], The probability threshold for pruning was 0.001 (divided by the number of nodes at this level to obtain an absolute threshold). (This is quite tolerant; in practice a larger threshold may be desirable to reduce the search.) Some preliminary crude pruning is included in this implementation. If the discrepancies from (3) are so large compared to the covariance matrix from (A) (4)-hat it is obvious that this node will be pruned, the further computations for this combination are omitted and thus this node does not appear in Figures 3 and A. No final adjustment at the bottom level of the search tree was done.

Figure 3 shows the results of using the data in Figure 1 as Scene A and the data in Figure 2 as Scene B. The search tree is shown. The numbers at the left followed by colons are the feature numbers In Scene A. The other numbers on

Figure 1. Ellipsoids found with 5-cm height threshold.

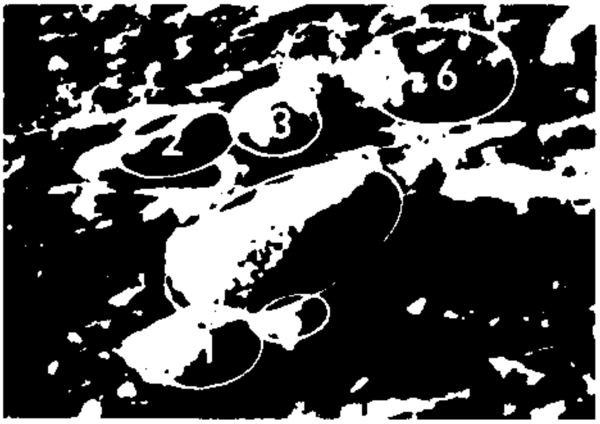
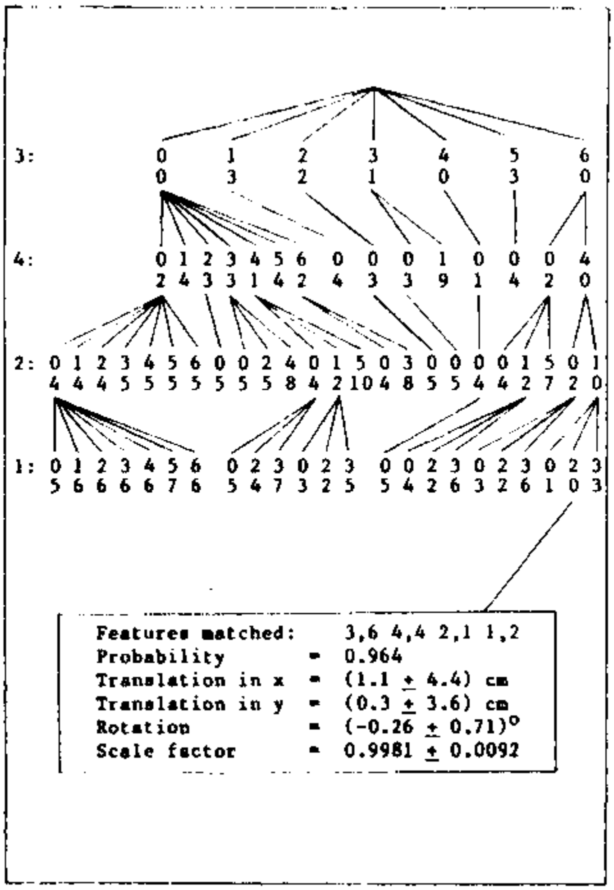
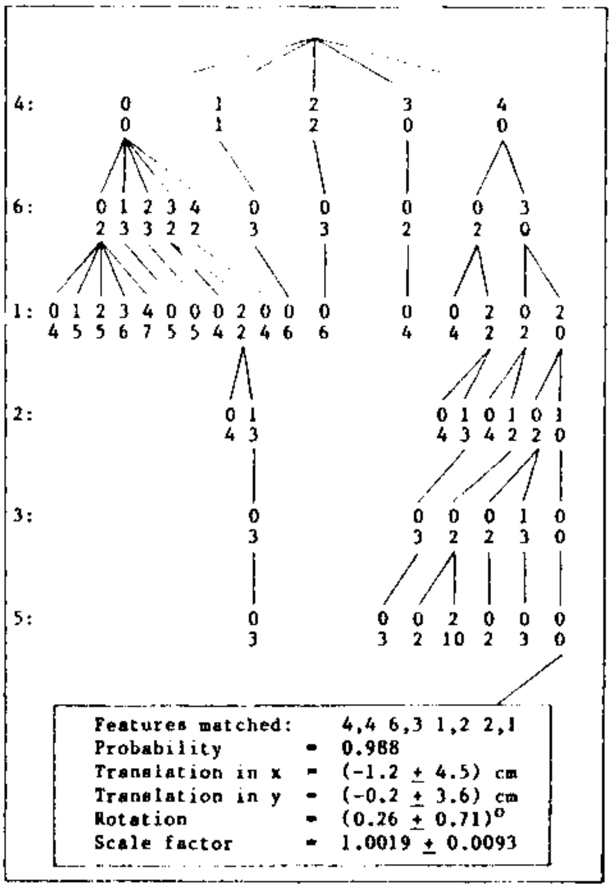


Figure 2. Ellipsoids found with 3-cm height threshold.



```
Features matched:      3,6 4,4 2,1 1,2
Probability          = 0.964
Translation in x     = (1.1 ± 4.4) cm
Translation in y     = (0.3 ± 3.6) cm
Rotation             = (-0.26 ± 0.71)°
Scale factor         = 0.9981 ± 0.0092
```

Figure 3. Match of scene in Fig. 1 to Scene in Fig. 2.



```
Features matched:      4,4 6,3 1,2 2,1
Probability          = 0.988
Translation in x     = (-1.2 ± 4.5) cm
Translation in y     = (-0.2 ± 3.6) cm
Rotation             = (0.26 ± 0.71)°
Scale factor         = 1.0019 ± 0.0093
```

Figure 4. Match of scene in Fig. 2 to scene in Fig. 1.

the same line are the feature numbers in Scene B for the features being matched to this feature in Scene A. Zero means that this feature in Scene A is left unmatched. The numbers just below the Scene B feature numbers represent the a posteriori probabilities computed for this match so far. To save space the. negative of the common logarithm of the probability, truncated to an Integer, is shown. Below the search tree the final results are shown for the most probable match, with computed standard deviations. Note that the probability is close to 1, whereas the a priori probability was only 1/2 This illustrates the fact that, when the features contain enough information to determine the match well, the a priori probabilities have little effect, as long as they are reasonable. Also note that the standard deviation of the scale factor is not much less than the input value of 0.01, which means that the solution was not able to add much information about the scale factor. (Since the two scenes were both from the same actual scene and same camera position, the true values of translation and rotation are zero, and the true value of scale factor is unity.)

Figure 4 similarly shows the results of doing the match with the scenes interchanged, so that Figure 2 now represents Scene A and Figure 1 represents Scene B. Even though the search tree is quite different, the final results are almost the same. (The two final results would agree even better if a complete solution were done at the bottom level of the search tree, since the same features were matched.) Of course, since the scenes have been interchanged, the translation and rotation have changed sign and the scale factor has been inverted.

In this example, only the two horizontal translation parameters were free to be adjusted by large amounts. Therefore, after the first level, the search tree stopped growing rapidly, because one feature match in two dimensions suffices to determine two translation parameters. If the a priori values of rotation and scale factor had been very uncertain also, the rapid growth would have continued to the second level, because two features matches in two dimensions are required to determine four parameters. However, the rate of growth in general would be less than exponential because of the decreasing computed probabilities of combined incorrect matches, based on local evidence (here only size). If more information (perhaps shape) had been included in the feature vector, the growth might have been less rapid, because better discrimination against incorrect individual feature matches would have been possible. Also, a higher probability threshold would reduce the growth. It can be seen that in Figure 3 three of the seven matches at the first level have a probability less than 0.01 and In Figure 4 one of the five does.

After the level of rapid growth discussed above, there is still some growth in the search tree caused by uncertainty in the position of the features. However, most of the growth at these level is caused by leaving the features in Scene A unmatched (indicated by zeros for feature numbers in Scene B). As more objects are left erroneously unmatched, the probability of these combinations becomes quite small at the lower levels of the tree, and they eventually are pruned.

## REFERENCES

[l] Fischler, M.A. and Elschlager, R.A. "The Representation and Matching of Pictorial Structures." IEEE Trans. Comput. C-22 (1973) 67-92.

[2] Gennery, D. B. "Modelling the Environment of an Exploring Vehicle by Means of Stereo Vision." AIM-339, STAN-CS-80-805, Computer Science Dept., Stanford University, June, 198a

[3] Brooks, R. A., Creiner, R., and Binford, T.O. "The ACRONYM Model-Based Vision System." In Proc. IJCAI-79. Tokyo, August, 1979, 105-113.

[4] Gennery, D.B. "Object Detection and Measurement Using Stereo Vision." In Proc. IJCAI-79. Tokyo, August, 1979, 320-327.

[5] Price, K. "Symbolic Matching and Analysis with Substantial Changes in Orientation." In Proc. Image Understanding Workshop (DARPA), Cambridge, MA," May, 1978, 93-99.

[6] Milgram, D. L. and Bjorklund, C. M. "Superposition", Lockheed Missiles and Space Company internal memo, Palo Alto, CA, 1979.

[7] Hohn, F. E. Elementary Matrix Algebra (Third Edition). The MacMillan Company, 1973.

[8] Mikhail, E. M. (with contributions by F. Ackermann). Observations and Least Squares. Harper & Row, 1976.

[9] Cooper, D. B. and Yabalik, N. "On the Computational Cost of Approximating and Recognizing Noise-Perturbed Straight Lines and Quadratic Arcs in the Plane." IEEE Trans. Comput. C-25 (1976) 1020-1032.