

# Optimal Geometric Model Matching Under Full 3D Perspective

J. Ross Beveridge  
Colorado State University  
Fort Collins, Colorado

Edward M. Riseman  
University of Massachusetts  
Amherst, Massachusetts

## Abstract

*Matching algorithms use random-start local search and a 3D pose recovery algorithm to find optimal matches between 3D object models and 2D image features. An algorithm using only a weak-perspective approximation to full 3D perspective solves a subset of the test problems presented. A second algorithm always uses an iterative 3D pose algorithm to account for 3D perspective and solves all test problems including those with varying 3D perspective. A third hybrid algorithm uses weak-perspective to direct search and 3D pose to periodically correct for perspective. It is faster than the second. A fourth algorithm is a hybrid which also uses a technique called ‘subset-convergence’ to escape from some local optima. It performs best on the most difficult matching problems.*

## 1 Introduction

Most work on matching three-dimensional (3D) geometric object models to features extracted from two-dimensional (2D) images neglects 3D perspective. More precisely, matching typically fails to quantitatively account for changes in object pose and associated projection of the model into the image. Instead, it is typical to require restrictions on the ways the appearance of an object can vary in an image. For instance, many matching techniques rely upon geometric constraints implied by small numbers of matched geometric features. This is a problem because these constraints generally do not hold under perspective.

Our work instead emphasizes optimization and the use of optimization algorithms based upon local search: a well established technique for solving difficult combinatorial optimization problems [Pap82]. Our algorithms dynamically best-fit object models to image features for competing matches, and consequently our original algorithms [Bev89, Bev90] now

have practical generalizations which handle perspective [Bev92a, Bev92b, Bev92]. A detailed account of all this work may be found in Beveridge’s Ph.D. dissertation [Bev93].

To handle perspective, the closed-form 2D fitting techniques presented in [Bev90] are replaced or supplemented with an iterative non-linear algorithm developed by Kumar [Kum89, Kum92, Kum93]. Kumar’s algorithm brings the projection of a 3D model into alignment with corresponding image features and thereby finds the best-fit 3D *pose* (position and orientation) of the object model relative to the camera. These algorithms make it possible to solve previously unsolvable recognition problems arising in the context of indoor robot navigation. This paper compares four distinct local search matching algorithms applied to landmark recognition problems.

## 2 Previous Work and 3D Perspective

An important distinction between our work and that which has come before concerns the way 3D features on a model are assumed to map to the 2D image plane. This mapping will here be called *imaging*. Most prior work has replaced *full-perspective* (a pin-hole camera model) imaging with less general imaging models. A large body of work uses 2D models or 2D projections of 3D models. When there is a 3D model, imaging is treated as a two step process with the object model projected into the image once from a particular pose and all subsequent changes made by subjecting the 2D projection to 2D affine transformations. Using a four parameter affine - rotation, translation and scale - a weakened form of perspective imaging called here *weak-perspective-2D* results. Fixing scale yields *2D-rigid* imaging. Table 1 describes conditions conducive to 2D-rigid and weak-perspective-2D imaging along with examples of previous work to use each.

Matching algorithms using weak-perspective-2D and 2D-rigid cannot adjust the appearance of the object model to account for many potential changes in object pose. To partially overcome this limitation, scaled-orthographic projection is sometimes used

---

\*This work was supported by the Defense Advanced Research Projects Agency (via TACOM) under contract DAAE07-91-C-R035 and by the National Science Foundation under grant CDA-8922572.

to map between 3D object features and the 2D image. This form of imaging will be here called *weak-perspective-3D*. Because the pose of the 3D object may be changed using weak-perspective-3D it can handle wide variations in pose. However, it does not induce vanishing points and will distort the appearance of objects with points differing significantly in depth. A sampling of researchers to use weak-perspective-3D are listed in Table 1.

To the best of our knowledge, only ourselves and Lowe have demonstrated matching with full-perspective imaging. There are perhaps two reasons for this. First, determining the pose of an object under full-perspective is not trivial. Second, common approaches rely upon local geometric constraints associated with sets of corresponding features. These sets must be larger for full-perspective than for the simpler imaging models. For example, in geometric hashing [Lam88], 2 points are required to establish a basis under weak-perspective-2D imaging while full-perspective requires 5 points.

### 3 Local Search Matching

Local search refers to a broad class of combinatorial optimization techniques which iteratively search a locally defined neighborhood until they arrive at locally optimal solutions [Pap82]. Multiple independent random trials are often used to increase the odds of seeing a near optimal solution. Different algorithms may use different local neighborhoods, employ different strategies for searching the neighborhood, and use different ways of evaluating competing solutions.

Four specific local search matching algorithms tailored to model matching are described below and their performance is analyzed on a series of hallway landmark recognition problems. The analysis exemplifies how a set of example problems can be used to characterize algorithm performance. All the algorithms tested here minimize a ‘fit plus omission’ match error described in the following section.

#### 3.1 Minimize Fit plus Omission

To compensate for errors in image feature extraction, many-to-many mappings between model and image features are supported. Hence, the space  $C$  of possible matches is the powerset of all possibly corresponding pairs of model segments  $m \in M$  and data segments  $d \in D$ .

$$C = 2^S \quad S \subseteq M \times D \quad (1)$$

A match error function is defined over  $C$ , and the optimal match  $c^*$  minimizes this error function:

$$E_{\text{match}}(c^*) \leq E_{\text{match}}(c) \quad \forall c \in C \quad (2)$$

| Imaging   | Used by ...   |
|---|---|
| 2D-rigid  |   |
| Flat objects perpendicular to the camera. The distance from the camera is known.  | Kalvin [Kal86]<br>Grimson [Gri87]<br>Stockman [Sto87]<br>Hwang [Hwa89]<br>Beveridge [Bev89]                   |
| Weak-perspective-2D   |   |
| Flat objects perpendicular to the camera but at unknown distance and placement relative to the camera. Also useful for general 3D objects given further restrictions upon possible viewpoint. | Ayache [Aya86]<br>Gottschalk [Got89]<br>Costa [Cos90]<br>Grimson [Gri90]<br>Beveridge [Bev90]<br>Cass [Cas92] |
| Weak-perspective-3D   |   |
| Any 3D object shallow in depth and viewed from any arbitrary viewpoint. Some distortion induced for all but perfectly flat objects.   | Lowe [Low85]<br>Thompson [Tho87]<br>Huttenlocher [Hut90]  |
| Full-perspective  |   |
| Any 3D object viewed from any arbitrary viewpoint. An initial approximate pose estimate is required.  | Lowe [Low91]<br>Beveridge [Bev92b]  |

Table 1: Previous work by imaging model.

The match error,  $E_{\text{match}}$ , is a combination of two terms.

$$E_{\text{match}}(c) = \left( \frac{1}{\sigma^2} \right) E_{\text{fit}}(c) + E_{\text{om}}(c) \quad (3)$$

$E_{\text{fit}}(c)$  is a residual squared-error obtained by fitting the model to the corresponding data.  $E_{\text{om}}$  penalizes matches which omit portions of the model from the match. The weighting coefficient  $\sigma$  controls the relative importance of these factors. The omission error  $E_{\text{om}}$  is defined to be a non-linear function of the fraction of the model line segments not covered by data line segments.

##### 3.1.1 Weak-perspective Fitting

The fit error  $E_{\text{fit}}(c)$  used in weak-perspective matching is a normalized function of the *integrated squared perpendicular distance* (ISPD) between data segments and infinitely extended model lines. More precisely, the 2D model is rotated, translated and scaled in the

image plane so as to minimize the sum of ISPD between corresponding segments and the residual ISPD is normalized to produce  $E_{\text{fit}}(c)$ .

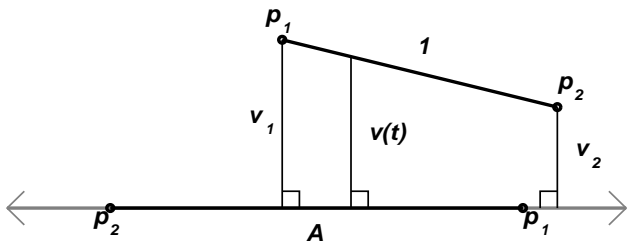


Figure 1: Points on data segment 1 project perpendicularly onto model line  $A$ . The perpendicular distance at any point along 1 may be written as a function  $v(t)$ .

The ISPD between a pair of segments may be written solely in terms of the perpendicular distance to endpoints. To see this, observe in Figure 1 how the distance  $v$  between a model line  $A$  and data segment 1 may be written as a function of parameter  $t$ :

$$v(t) = v_1 + (v_2 - v_1) \frac{t}{\ell} \quad 0 \leq t \leq \ell. \quad (4)$$

The definite integral of  $v(t)^2$  over the length of the data  $\ell$  segment has a relatively simple form:

$$\text{ISPD} = \int_0^\ell v^2(t) dt = \frac{\ell}{3} (v_1^2 + v_1 v_2 + v_2^2). \quad (5)$$

### 3.1.2 Object Pose Under 3D Perspective

The basis for Kumar's algorithm [Kum89a] lies in the realization that, in the absence of errors, a model line in 3-space lies in a plane defined by the projection of this line in the image plane and the focal point of the camera. This relation is illustrated in Figure 2. The origin of the camera coordinate system is the camera focal point, and the focal point plus the two endpoints of an image line segment define a plane in 3-space. If the image line segment is the projection of a line in the 3D world, then this 3D world line segment *must* lie on this plane. Due to noise, the segment usually will not lie exactly in the plane, but the distance from the 3D world segment to the plane should be small. Kumar's pose algorithm solves for a rigid transformation which minimizes the sum-of-squared distances between points on 3D world lines and these planes. The method for solving this nonlinear least-squares optimization algorithm is adapted from Horn [Hor90], and in particular uses quaternions to represent rotation in order to speed convergence. Several extensions to Kumar's algorithm which make it better for matching are described in Beveridge's Ph.D. dissertation [Bev93].

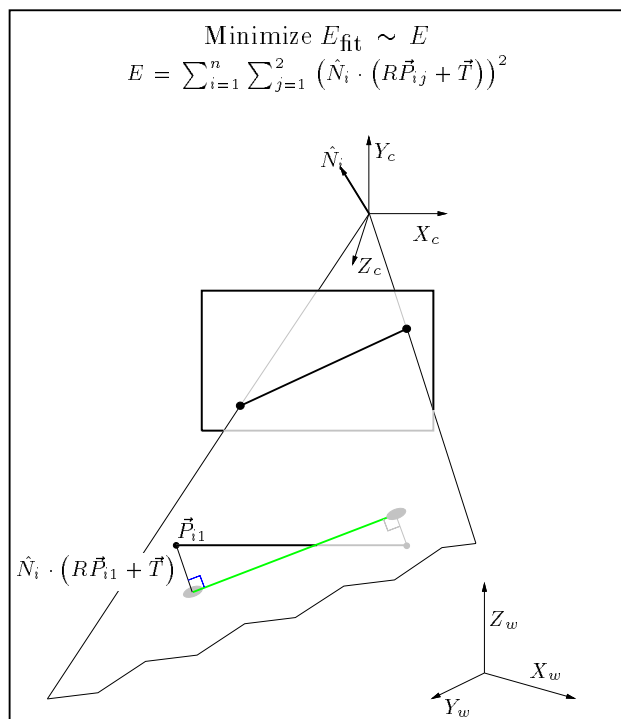


Figure 2: Point-to-plane fit for 3D pose algorithm.

## 3.2 Hallway Navigation

Perspective must be accounted for to solve some indoor landmark-based robot navigation problems. For example, a robot moving through a hallway will experience dramatic perspective effects associated with even modest movements. This is illustrated in Figure 3. Figure 3a shows 9 different poses for a robot. A partial wire frame model of the hallway is shown in relation to these poses. Figure 3b shows the prominent model or landmark features as they would appear from each of the 9 indicated poses.

The partial hallway model, the 9 pose estimates, and the two images shown in Figures 4a and 4b form the basis for the experiments presented below. The goal of the experiments in Section 3.4.1 is to recover the true pose that produced the image of Figure 4a; each of the 9 pose estimates shown in Figure 3 will be used as initial estimates of robot position. The goal of the experiment in Section 3.4.2 is to recover the pose from which each of the two images in Figure 4 were taken given the other pose as an initial estimate.

## 3.3 Four Matching Algorithms

Four local search matching algorithms are presented in this section. The first is a weak-perspective algorithm and the remaining three are

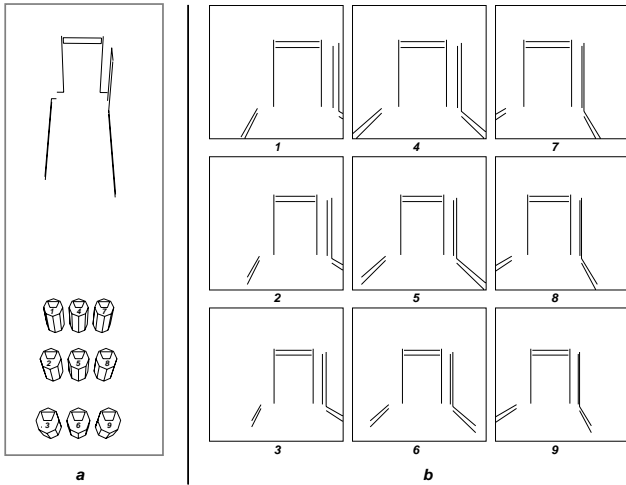


Figure 3: Perspective views of landmarks: a) Robot in relation to a partial hallway model. b) Landmark features as they would appear from each of the 9 poses.

full-perspective algorithms, meaning they use Kumar’s pose algorithm to vary 3D perspective during matching. Table 2 summarizes the major attributes of the four algorithms. Each algorithm is briefly described below and a fuller description may be found in Beveridge’s Ph.D. dissertation [Bev93]. Additionally, the weak-perspective-steepest-descent algorithm appears in [Bev90], the full-perspective-inertial-descent algorithm in [Bev92a], the hybrid-weak-full-perspective-algorithm in [Bev92b], and the hybrid-subset-convergent algorithm in [Bev92].

The diagram shows a grid of lines representing a hallway structure. The table below it shows matching results for four different algorithms. The table has 4 rows and 4 columns. The first row is empty. The second row has 'x' marks in the second, third, and fourth columns. The third row has 'x' marks in the second, third, and fourth columns. The fourth row has 'x' marks in the second, third, and fourth columns. The fifth row has 'x' marks in the second, third, and fourth columns. The sixth row has 'x' marks in the second, third, and fourth columns. The seventh row has 'x' marks in the second, third, and fourth columns. The eighth row has 'x' marks in the second, third, and fourth columns. The ninth row has 'x' marks in the second, third, and fourth columns. The tenth row has 'x' marks in the second, third, and fourth columns.

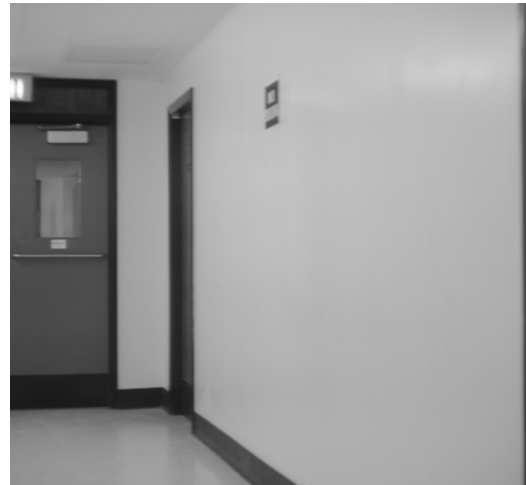
|  |  |   |   |   |
|--|--|---|---|---|
|  |  |   |   |   |
|  |  | x | x | x |
|  |  | x | x | x |
|  |  | x | x | x |
|  |  | x | x | x |
|  |  | x | x | x |
|  |  | x | x | x |
|  |  | x | x | x |
|  |  | x | x | x |
|  |  | x | x | x |
|  |  | x | x | x |

Table 2: Comparing matching algorithm attributes.

### 3.3.1 Weak-perspective Steepest-Descent



(a)



(b)

Figure 4: Hallway images. a) Image 1, b) Image 2.

This and subsequent algorithms use a local neighborhood defined as all correspondences  $c'$  obtained by adding or removing a single pair  $s \in S$  from the current correspondence  $c$ . Figure 5 illustrates the search of such a neighborhood using a steepest-descent strategy. Traces of two examples of local search are indicated by the rows in the two tables in Figure 5. A filled-in square indicates a pair of model-data segments are included in a match. The first row in each table indicates a randomly selected correspondence and the final row one which is locally optimal. Successive rows differ by only one pair of segments. This is called the Hamming-distance-1 neighborhood because when correspondences are represented as bit strings the neighborhood consists of all strings within Hamming distance 1 of the current string.

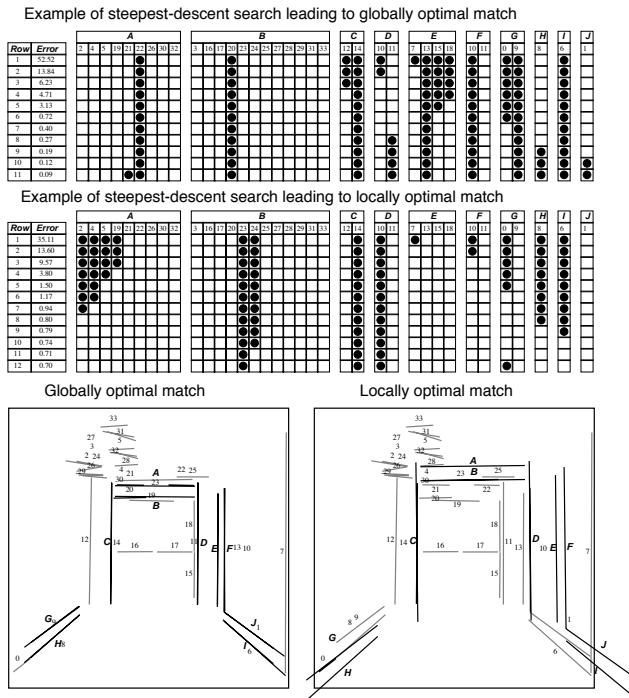


Figure 5: Example of steepest-descent local search using weak-perspective.

Search moves from the initial match to successively better matches by considering *all* neighboring matches and moving to the one which is best. Using the ISDP measure defined earlier, the model is fit to the data for each neighbor and the associated match error is determined. The matching problem illustrated is for the hallway model as projected from pose 2 and a subset of the straight line segments extracted using the Burns algorithm [Bur86] from the image shown in Figure 4a. In this case weak-perspective is adequate to reasonably fit the projected model to the data. The top table shows a trace of local search which terminates with the globally optimal match illustrated on the bottom left of Figure 5. The lower table shows a trace terminating with the locally but not globally optimal match shown in the lower right.

With all our local search algorithm, multiple trials are run to increase the odds of finding the optimal match. When the initial correspondences are selected independently and at random, the probability of failing to find the globally optimal match over a series of trials is conjunctive. Thus, even if the probability of success on a single trial,  $P_s$ , is fairly low, a modest number of trials is generally sufficient to find the globally optimal match with high confidence. The probability of success  $P_s$  is in some sense a measure of an algorithm's performance and Section 3.4 will begin the

comparison of different algorithms by estimating  $P_s$  for each. When the weak-perspective-steepest-descent algorithm is used for landmark based navigation, the 3D pose algorithm is used at the end of matching to infer robot position from the best weak-perspective match.

### 3.3.2 Full-perspective-Inertial-Descent

The full-perspective-inertial-descent algorithm uses a modified version of the match error in which the fit error  $E_{fit}$  is defined to be a function of the residual point-to-plane squared error terms used by the 3D pose algorithm and illustrated in Figure 2. This means the 3D pose of the object model relative to the camera is computed for *every* match which is tested, i.e. each of the  $n$  neighbors in the Hamming-Distance-1 neighborhood.

Evaluating the  $n$  neighboring matches using the 3D pose algorithm is computationally demanding. Using steepest-descent,  $n$  3D poses would be computed before every move. To save computation, a modified strategy named here *inertial-descent* attempts to find a sequence of moves each time the neighborhood is evaluated. To do this, when the  $n$  neighbors of a current match are evaluated, those neighborhood transformations which lead to better matches are ranked in order of improvement and stored in a list. Provided this list is not empty, the algorithm applies the first transformation and thus moves to a guaranteed better match state. Then the algorithm tests whether the second ranked transformation applied to this new match generates an even better match. If it does, it repeats this process until either the list of transformations is exhausted or a transformation is encountered which no longer yields improvement, and then all  $n$  neighbors are tested and a new list built.

To emphasize that new 3D poses are generated during search, Figure 6 shows the projection of the landmark from intermediate matches for one execution of the inertial-descent algorithm. In this example the landmark is being matched to data extracted from the image shown in Figure 4b. Since the 3D pose algorithm is used by this and the following two algorithms, these algorithm determine the position of the robot relative to the landmarks when used for landmark recognition.

### 3.3.3 Hybrid Weak- and Full-perspective

Fitting the projection of the landmark using the closed-form weak-perspective algorithm takes roughly an order of magnitude less computation than computing 3D pose using Kumar's algorithm. For efficiency, the hybrid-weak-full-perspective algorithm is used because it employs the closed-form algorithm to

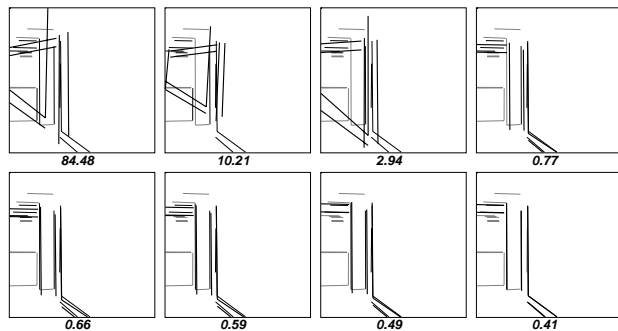


Figure 6: Landmark projections for full-perspective-inertial-descent example.

fit a projection of the landmark to each of the  $n$  neighboring matches. Only when a new match becomes the current match is the 3D pose recomputed and the model reprojected.

After reprojecting the model,  $E_{\text{match}}$  measured relative to the new 2D projection can actually increase. To avoid premature termination of search in these cases, search is permitted to continue for a fixed number of moves even if  $E_{\text{match}}$  is increasing. Up to 10 moves were permitted in the experiments reported here. Checks must be put in to prevent search from cycling back to the already discovered best match. A variant of local search called *tabu search* [Glo89] embodies this idea: moves which in some sense lead backwards are taboo (Glover uses spelling 'tabu'). For the hybrid-weak-full-perspective a move is taboo if it adds or removes a pair of model-data features already added or removed since the last best match was found.

### 3.3.4 Hybrid Subset-Convergent

We've developed a strategy for getting out of local optima which we call subset-convergent local search. The underlying idea is to test whether subsets of a locally optimal match in the Hamming-distance-1 neighborhood are 'consistent' with the overall match, and more importantly, by testing subsets to escape from poor quality locally optimal matches.

The hybrid-subset-convergent algorithm searches using the hybrid-weak-full-perspective algorithm until it finds a local optima using this algorithm. It then initiates search using this same algorithm from subsets of this locally optimal match. These subsets include model-data pairs associated with selected model line segments. In keeping with all our past work, only *four* preselected subsets of model segments are used. These subsets are selected based by automated heuristic procedure which attempts to select pairs of model segments as subsets which are long and come close to meeting at a common point. In the experiments which

follow, the pairs of segments  $(A, D)$ ,  $(B, C)$ ,  $(F, I)$  and  $(E, J)$  as labeled in Figure 5 are used.

Because search is initiated from subsets only after matches locally optimal using the hybrid-weak-full-perspective algorithm are found, the hybrid-subset-convergent algorithm will always do as well or better than the hybrid-weak-full-perspective algorithm on a single trial. A full description of the subset-convergent control strategy for local search appears in Beveridge's Ph.D. dissertation [Bev93].

## 3.4 Comparing Algorithm Performance

This comparison is divided into two experiments. The first involves recognizing the hallway landmark segments in Image 1 from each of the 9 initial pose estimates shown in Figure 3a. A second experiment involves a greater error in the initial pose estimate.

### 3.4.1 Experiment 1: Modest Pose Errors.

Each of the algorithms introduced above is tested on the task of recovering the true pose or our robot, pose 5, from each of the estimates shown in Figure 3a. The true pose is 41.3 feet from the doorway and 4 feet from each of the two side walls. The estimates are obtained by introducing translation errors of 4 feet forward and backward and 2 feet side-to-side.

In establishing the search spaces for each of the landmark matching problems, the predicted appearances of the landmarks, shown in Figure 3b, are used to determine the sets of candidate pairs. A model-data pair  $s = (m, d) \in M \times D$  is an element of set  $S$  if:

- 1)  $d$  is within 30 degrees of  $m$ .
- 2)  $d$  is within 128 pixels of  $m$ .
- 3)  $d$  is at least 1/4 the length of  $m$ .
- 4)  $d$  and  $m$  have the same sign-of-contrast.

The first test filters candidate pairs based upon relative orientation. The second is a rough proximity test; the image is 512 pixels across, so the threshold of 128 pixels represents one quarter of the distance across the image. The third test removes excessively small fragments from consideration. The thresholds for these three tests are picked based upon experience with the domain, and are adequate to insure that the correct match is contained in the resultant search space.

The fourth constraint, sign-of-contrast, is useful in a domain such as the hallway, where many of the segments are the result of surface markings rather than occlusion. For example, it is known with certainty that the top edge of the baseboards in the hallway will be light on top and dark on the bottom, and therefore straight line segments extracted from the image with

the opposite contrast need not be considered. However, sign-of-contrast is not a reliable constraint for occlusion edges where the background surfaces will vary. To test the importance of the sign-of-contrast constraint in helping to solve these landmark matching problems, all the matching experiments in this section will be conducted using candidate pairs obtained with and without this constraint. The set  $S$  denotes pairs filtered on contrast and set  $\bar{S}$  not filtered on contrast. Typically, ignoring sign-of-contrast doubles the number of pairs, and Table 3 shows the exact size of these sets for all 9 initial pose estimates.

|             | Initial Pose Estimate |    |     |    |    |    |    |    |     |
|-------------|-----------------------|----|-----|----|----|----|----|----|-----|
|             | 1                     | 2  | 3   | 4  | 5  | 6  | 7  | 8  | 9   |
| $ S $       | 41                    | 45 | 53  | 36 | 37 | 43 | 42 | 42 | 54  |
| $ \bar{S} $ | 87                    | 94 | 112 | 75 | 77 | 92 | 89 | 94 | 112 |

Table 3: Size of candidate pair sets with/without the sign-of-contrast. Pairs  $S$  are tested for compatible sign-of-contrast, pairs  $\bar{S}$  are not.

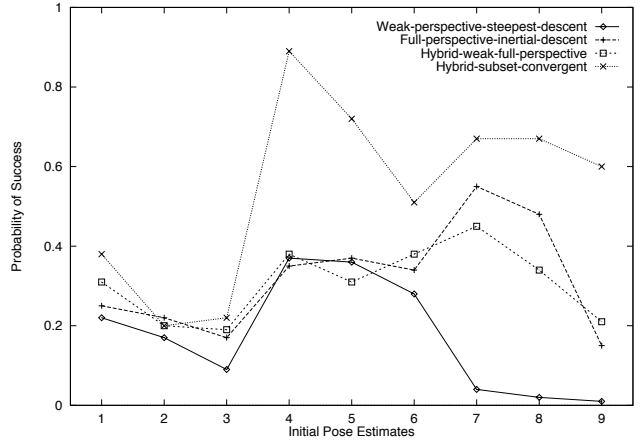
All three algorithms using the 3D pose algorithm during matching *recover the true pose* in all the experiments conducted for this section<sup>1</sup>. In fact, they all reliably find exactly the same optimal correspondence in the 18 matching problems resulting from the 9 initial pose estimates and the two sets of candidate pairs  $S$  and  $\bar{S}$ . Consequently, the robot’s true pose is recovered to within the accuracy bounds of our pose algorithm [Kum90]<sup>2</sup>. The weak-perspective-steepest-descent algorithm did equally well for initial pose estimates 4, 5 and 6, where the landmark projections could successfully be rotated, translated and scaled to reasonably fit the actual data. However, for the other cases the 3D pose estimates derived from the best matches differed from the true pose. They were off by 1 to 2 feet in 5 of the 6 cases, and by nearly 8 feet in one case.

Between 100 and 300 trials of random-start local search for each of the four algorithms was run on each of the 18 matching problems. Both the fractions of the runs finding the globally optimal match and the average run-time per trial were recorded for each algorithm applied to each problem. The maximum likelihood estimate of the probability of successfully finding the globally optimal match on a single trial  $\hat{P}_s$  is simply the ratio of the number of times the globally match was found over the total number of trials run. These

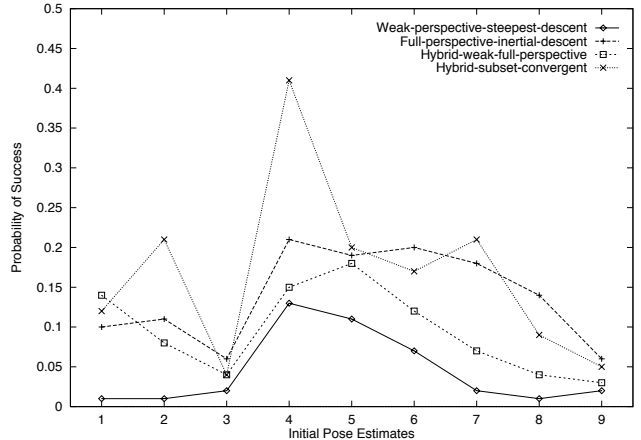
<sup>1</sup>Only the position portion of the pose estimate associated with a match is considered.

<sup>2</sup>Speaking loosely, for these problems the 3D pose algorithm appears accurate to within roughly 6 inches.

estimates are plotted in Figure 7. Figure 7a shows  $\hat{P}_s$  for each of the 9 pose estimates and candidate pairs  $S$  filtered by sign-of-contrast. Figure 7b shows  $\hat{P}_s$  for the corresponding 9 problems using candidate pairs  $\bar{S}$  in which the sign-of-contrast constraint is ignored.



(a)



(b)

Figure 7:  $\hat{P}_s$  for matching from 9 pose estimates. a) Candidate pairs filtered by sign-of-contrast, b) Candidate pairs not filtered by sign-of-contrast.

The  $\hat{P}_s$  values drop when sign-of-contrast is not taken into account. As measured by  $\hat{P}_s$ , matching problems not using the sign-of-contrast constraint are considerably more difficult to solve. A potential general explanation for the difference lies in size of the search spaces in the two cases. Without the sign-of-contrast constraint the set of candidate pairs roughly doubles in size, and the search space grows astronomically from about  $2^{50}$  to  $2^{100}$  states. There is another factor quite particular to this problem. Without the sign-of-contrast constraint, the upper and lower borders of the hallway baseboards are locally ambiguous.

The tops of the baseboard in the model can quite readily match the bottoms in the image, and vice versa.

Another thing to observe from the plots in Figure 7 is the relative performance of the four algorithms. Consistently in both cases, the hybrid-subset-convergent algorithm is nearly as good as or better than the full-perspective-inertial-descent algorithm. *explain that it must perform as well because it goes beyond local optima of hybrid.* As is to be expected, the hybrid-subset-convergent algorithm outperforms the hybrid-weak-full-perspective algorithm. Finally, the weak-perspective-steepest-descent algorithm typically does worse than the other three.

The next set of plots, Figures 8a and 8b, show the estimated run-time,  $\hat{r}_s$ , required to be confident of finding the globally optimal match. To compute  $\hat{r}_s$ , first the number of trials  $\hat{t}_s$  required to find the globally optimal match with probability 0.95 or better is computed using the following equation.

$$\hat{t}_s = \lceil \log_{(1-\hat{p}_s)} 0.05 \rceil. \quad (6)$$

This equation is derived from the fact that probability of failure to see the globally optimal match in a series of  $t$  independent trials is conjunctive (see Beveridge's Ph.D. dissertation [Bev93]).

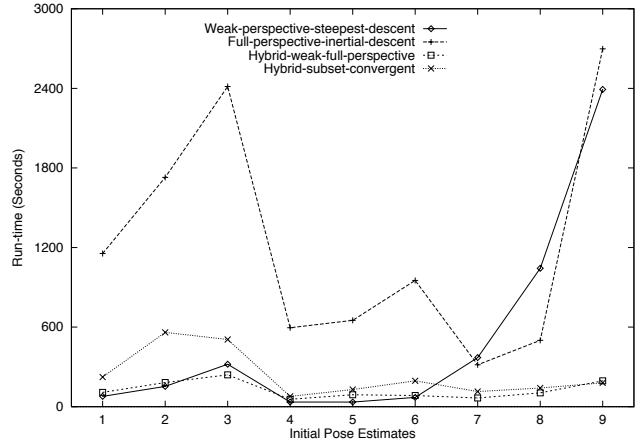
The estimated run-time  $\hat{r}_s$  is the average time per trial  $r$  multiplied by  $\hat{t}_s$ . The average run-times reported here are for a T. I. Explorer II Lisp Machine. As with the values for  $\hat{P}_s$ , perhaps the most striking difference is the increase in estimated run-time for the problems not utilizing the sign-of-contrast constraint. The range of the plot in Figure 8a is 50 minutes, while the range for the plot in Figure 8b is 5 hours. Again, it must be remembered that the search space in the latter case is incomparably larger.

Overall, the hybrid-weak-full-perspective algorithm seems to be outperforming the others. A new C version of this algorithm running on a Decstation 5000 is roughly 50 times faster than the Lisp Machine version, and using this version, the hybrid-weak-full-perspective algorithm solves these problems in between 1 to 5 seconds when sign-of-contrast is taken into account.

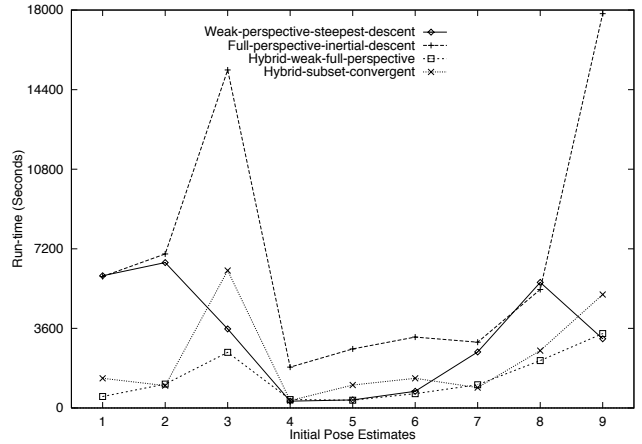
### 3.4.2 Experiment 2: Larger Pose Errors.

In this second experiment, the pose for image 1 shown in Figure 4a is given as the initial estimate of robot pose when the true pose is that for image 2 shown in Figure 4b, and vice versa. Both the landmark projections from the initial pose estimates, as well as the best matches, are shown for these two problems in Figure 9.

These are difficult matching problems. In one case



(a)



(b)

Figure 8: Estimated run-times for matching from 9 pose estimates (C version is roughly 50 times faster): a) Using sign-of-contrast constraint, b) Not using the sign-of-contrast constraint.

half the expected segments are not visible while in the other many unexpected segments are visible. Additionally, the proximity constraint for selecting candidate pairs  $S$  is relaxed to include all segments  $d$  within 256 pixels in order to account for the greater uncertainty in pose. For these problems, 300 trials are used to estimate  $\hat{P}_s$ .

The results are summarized in Table 4. Only the three algorithms capable of handling full-perspective were tested on these problems. Comparing the run-times  $\hat{r}_s$  using each algorithm, the hybrid-subset-convergent is superior to the other two.

A basic tradeoff in local search is whether to run a few expensive trials that are more likely to succeed or instead many cheaper trials that are less likely to succeed? The full-perspective-inertial-descent al-

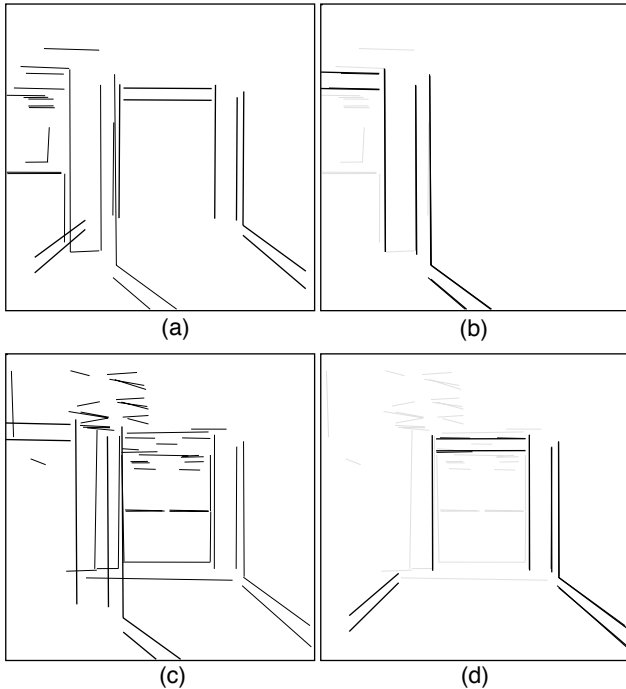


Figure 9: Confusing poses for images 1 and 2. a) landmark projection from pose 1 overlaying image 2 data, b) optimal match, c) landmark projection from pose 2 overlaying image 1 data. d) optimal match.

algorithm is slow but reliable, finding the globally optimal match with modestly high probability on each trial, but taken a long time to run each trial. In contrast, the hybrid-weak-and-full-perspective only finds the optimal match in 1 or 2 out of every 100 trials, but each trial runs in comparatively little time. In this particular case, the two factors balance, and the run-time  $\hat{r}_s$  is lower for the full-perspective-inertial-descent algorithm on one of the two matching problems, and higher on the other problem. Interestingly, the hybrid-subset-convergent lies in between these two extremes, and in terms of  $\hat{r}_s$  it out performs both.

## 4 Conclusion

Weak-perspective matching algorithms are inadequate for landmark-based robot navigation in a hallway, at least for the types of line-segment-based geometric matching problems studied here. The failure of the weak-perspective-steepest-descent algorithm has been demonstrated. These problems provided a powerful incentive to develop quantitatively the accurate full-perspective matching algorithms presented here.

The experiments with different full-perspective matching algorithms have shown that it is the hybrid

| Image 2 to Image 1                |             |     |             |             |
|-----------------------------------|-------------|-----|-------------|-------------|
| Algorithm.                        | $\hat{P}_s$ | $r$ | $\hat{t}_s$ | $\hat{r}_s$ |
| Full-perspective-inertial-descent | 0.29        | 172 | 9           | 1,548       |
| Hybrid-weak-full-perspective      | 0.02        | 6   | 149         | 894         |
| Hybrid-subset-convergent          | 0.09        | 25  | 32          | 800         |

| Image 1 to Image 2                |             |     |             |             |
|-----------------------------------|-------------|-----|-------------|-------------|
| Algorithm.                        | $\hat{P}_s$ | $r$ | $\hat{t}_s$ | $\hat{r}_s$ |
| Full-perspective-inertial-descent | 0.18        | 63  | 18          | 1,134       |
| Hybrid-weak-full-perspective      | 0.01        | 5   | 299         | 1,495       |
| Hybrid-subset-convergent          | 0.06        | 11  | 49          | 539         |

Table 4: Matching results when poses for Image 1 and Image 2 are confused. The probability of success  $\hat{P}_s$ , average run-time per trial  $r$ , required trials  $\hat{t}_s$ , and estimated run-time to solve the problem  $\hat{r}_s$  are shown for each algorithm applied to each matching problem. Times are reported in seconds.

algorithms, which blend the use of weak-perspective and full-perspective evaluation of matches *during* search, which perform best. They categorically beat the weak-perspective algorithm, which fails to find the proper match when the initial estimate of the 3D object’s pose introduces perspective distortion, and dramatically outperform the more brute-force full-perspective-inertial-descent algorithm in terms of speed. The hybrid-weak-full-perspective algorithm appears to perform somewhat better in terms of overall run-time for the modest pose error problems in Section 3.4.1, while the hybrid-subset-convergent version is more economical on the problems with larger initial pose error tested in Sections 3.4.2.

## References

- [Aya86] N. Ayache and O. D. Faugeras. Hyper: A new approach for the recognition and positioning of 2-d objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):44 – 54, January 1986.
- [Bev92] J. Ross Beveridge. Comparing subset-convergent and variable-depth local search on perspective sensitive landmark recognition problems. In *Proceedings: SPIE Intelligent Robots and Computer Vision XI: Algorithms, Techniques, and Active Vision*, volume 1825, pages 168 – 179. SPIE, November 1992.
- [Bev93] J. Ross Beveridge. *Local Search Algorithms for Geometric Object Recognition: Optimal Correspondence and Pose*. PhD thesis, University of Massachusetts at Amherst, May 1993.
- [Bev92a] J. Ross Beveridge and Edward M. Riseman. Can too much perspective spoil the view? a case study in 2D affine versus 3D perspective model matching. In *Proceedings: Image Understanding Workshop*,

- pages 665 – 663, San Mateo, CA, January 1992. Morgan Kaufmann.
- [Bev92b] J. Ross Beveridge and Edward M. Riseman. Hybrid weak-perspective and full-perspective matching. In *Proceedings: IEEE 1992 Computer Society Conference on Computer Vision and Pattern Recognition*, pages 432 – 438. IEEE Computer Society, June 1992.
- [Bev89] J. Ross Beveridge, Rich Weiss, and Edward M. Riseman. Optimization of 2-dimensional model matching. In *Proceedings: Image Understanding Workshop*, pages 815 – 830, Los Altos, CA, June 1989. DARPA, Morgan Kaufmann Publishers, Inc (Also a Tech. Report).
- [Bev90] J. Ross Beveridge, Rich Weiss, and Edward M. Riseman. Combinatorial optimization applied to variable scale 2D model matching. In *Proceedings of the IEEE International Conference on Pattern Recognition 1990, Atlantic City*, pages 18 – 23. IEEE, June 1990.
- [Bur86] J. B. Burns, A. R. Hanson, and E. M. Riseman. Extracting straight lines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-8(4):425 – 456, July 1986.
- [Cas92] Todd A. Cass. Polynomial-time object recognition in the presence of clutter, occlusion, and uncertainty. In *Proceedings: Image Understanding Workshop*, pages 693 – 704, San Mateo, CA, January 1992. DARPA, Morgan Kaufman.
- [Cos90] Mauro Costa, Robert M. Haralick, and Linda G. Shapiro. Optimal affine - invariant point matching. In *Proceedings of the IEEE International Conference on Pattern Recognition 1990, Atlantic City*, pages 233 – 236. IEEE, June 1990.
- [Glo89] F. Glover. Tabu search – part i. *ORSA Journal on Computing*, 1(3):190 – 206, 1989.
- [Gri90] W. Eric L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, Cambridge, MA, 1990.
- [Gri87] W. E. L. Grimson and T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(3):469–482, 1987.
- [Got89] Paul G. Gottschalk, Jerry L. Turney, and Trevor N. Mudge. Efficient recognition of partially visible objects using a logarithmic complexity matching technique. *International Journal of Robotics Research*, 8(6):110 – 131, December 1989.
- [Hor90] B. K. P. Horn. Relative orientation. *International Journal of Computer Vision*, 4:59 – 78, 1990.
- [Hut90] Daniel P. Huttenlocher and Shimon Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195 – 212, November 1990.
- [Hwa89] Vincent S. S. Hwang. Recognizing and locating partially occluded 2-d objects: Symbolic clustering method. *IEEE Trans. on Syst., Man, Cybern.*, 19(6):1644 – 1656, November 1989.
- [Kum90] Rakesh Kumar and Allen Hanson. Analysis of different robust methods for pose refinement. In *Proc. of IEEE Workshop on Robust Methods in Computer Vision*, pages 161 – 182, Seattle, WA, 1990. IEEE.
- [Kum93] Rakesh Kumar and Allen R. Hanson. Robust methods for estimating pose and a sensitivity analysis. *CVGIP:Image Understanding*, ?:(to appear), 1993.
- [Kum92] Rakesh Kumar. *Model Dependent Inference of 3D Information From a Sequence of 2D Images*. PhD thesis, University of Massachusetts, Amherst, February 1992.
- [Kum89a] Rakesh Kumar and Allen Hanson. Robust estimation of camera location and orientation from noisy data having outliers. In *Proc. of IEEE Workshop on Interpretation of 3D Scenes*, pages 52 – 60, Austin, TX, 1989. IEEE.
- [Kum89] Rakesh Kumar. Determination of camera location and orientation. In *Proceedings: Image Understanding Workshop*, pages 870 – 881, Los Altos, CA, June 1989. DARPA, Morgan Kaufmann Publishers, Inc.
- [Kal86] Alan Kalvin, Edith Schonberg, Jacob T. Schwartz, and Micha Sharir. Two-dimensional, model-based, boundary matching using footprints. *The International Journal of Robotics Research*, 5(4):38 – 55, 1986.
- [Low85] David G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, 1985.
- [Low91] David G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441 – 450, May 1991.
- [Lam88] Y. Lamdan and H. J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proc. IEEE Second Int. Conf. on Computer Vision*, pages 238 – 249, Tampa, December 1988.
- [Pap82] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*, chapter Local Search, pages 454 – 480. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [Sto87] George Stockman. Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, 40:361 – 387, 1987.
- [Tho87] D. W. Thompson and J. Mundy. Three-dimensional model matching from an unconstrained viewpoint. In *Proc. IEEE International Conference on Robotics and Automation*, pages 208 – 220, 1987.