

# Hybrid Weak-Perspective and Full-Perspective Matching.

J. Ross Beveridge      E. M. Riseman

Computer and Information Science Department  
University of Massachusetts at Amherst  
Amherst, MA. 01003 \*†

## Abstract

Full-perspective mappings between 3D objects and 2D images are more complicated than weak-perspective mappings, which consider only rotation, translation and scaling. Therefore, in 3D model-based robot navigation, it is important to understand how and when full-perspective must be taken into account. In this paper we use a probabilistic combinatorial optimization algorithm to search for an optimal match between 3D landmark and 2D image features. Three variations are considered: a weak-perspective algorithm rotates, translates and scales an initial 2D projection of the 3D landmark. A full-perspective algorithm always recomputes the robot's pose and reprojects the landmark when testing alternative matches. Finally, a hybrid algorithm uses weak-perspective to select a most promising alternative, but then updates the pose and reprojects the landmark. The hybrid algorithm appears to combine the best attributes of the other two. Like the full-perspective algorithm, it reliably recovers the true pose of the robot, and like the weak-perspective algorithm, it runs 5 to 10 faster than the full-perspective algorithm.

## 1 Introduction

The problem of matching 3D models to 2D image features arises in many domains. In this paper we consider problems associated with robot navigation. For example, a robot moving through a hallway tracks its progress using vision, and it acquires images such as shown in Figures 1 and 2. It must test and update its position estimate based upon the appearance of known landmarks.

---

\*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.

†Appeared in Proceedings: IEEE 1992 Computer Society Conference on Computer Vision and Pattern Recognition, pg 432 - 438.



Figure 1: Image 1

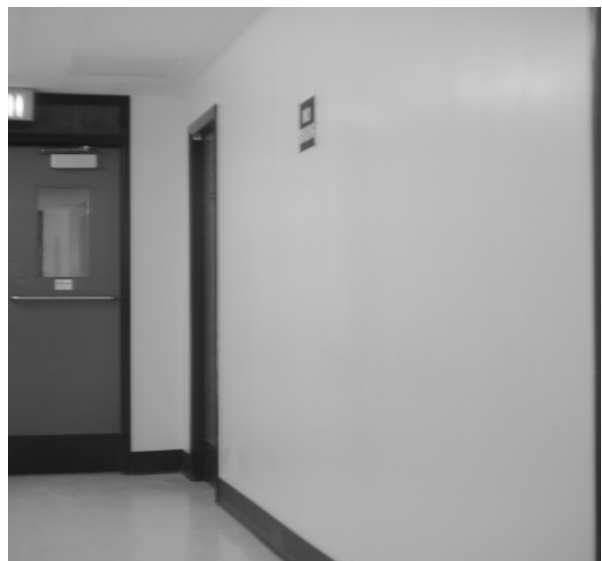


Figure 2: Image 2

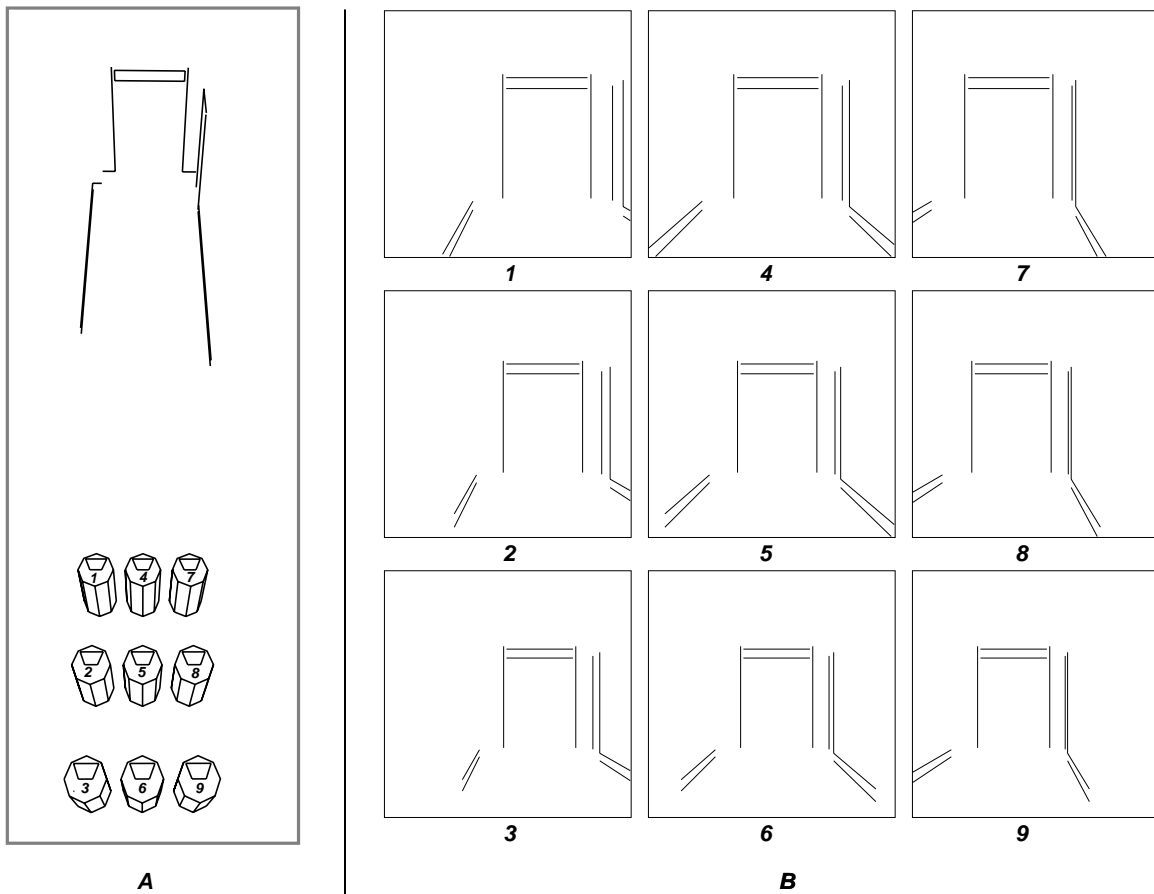


Figure 3: Perspective views of landmarks: (a) Robot in relation to a partial model of the hallway. The image in Figure 1 was taken from pose 5. Each of the 9 poses shown are used as initial estimates from which landmark recognition tries to recover the robot’s true pose. (b) Landmark features as they would appear from each of the 9 poses. The relative orientation of the baseboards and the door frame change as the robot’s pose estimate shifts laterally.

As illustrated in Figure 3, modest changes in position can introduce considerable perspective effects. Figure 3a shows 9 different poses – positions and orientations – for a robot. A partial wire frame model of the hallway is shown in relation to these poses. Figure 3b shows the prominent landmark features as they would appear from each of the 9 indicated poses. Landmark recognition involves matching prominent landmark features with features extracted from images (for example Figures 1 and 2). Image features used in this paper are 2D line segments produced by the Burns algorithm [Bur86].

How and when during matching should full-perspective projection be taken into account? Three alternatives are:

**Weak-perspective:** Match projected landmark features to image features subject to rotation, trans-

lation and scaling in the image plane.

**Full-perspective:** Match 3D landmark features to image features, updating pose and reprojecting landmark features for *every* correspondence tested during search.

**Hybrid:** Match 3D landmark features to image features. Use weak-perspective to rank alternatives correspondences. Use full-perspective to update pose and reproject the landmark after selecting an alternative, improved, correspondence.

In all three cases, the resultant match is used to estimate the robot’s true pose using an algorithm developed by R. Kumar [Kum89, Kum90]. This same algorithm is also employed by the full-perspective and hybrid algorithms to update pose during matching.

Each of these three algorithms will be evaluated on: 1) how well the true pose is recovered, and 2) how much computation is required. Results for the weak-perspective and full-perspective algorithms have previously been reported in [Bev92]. The hybrid-perspective algorithm is presented here for the first time.

## 2 Landmark Matching

The problem of matching landmark features is cast in terms of combinatorial optimization. This section defines the combinatorial space of correspondences, the optimality criterion, and the search algorithms used to probabilistically find optimal matches between 3D landmark features and 2D image features.

### 2.1 Correspondence Space

Let  $M$  be the set of 3D line segments in the landmark model, let  $D$  be the set of 2D data line segments in the image, and let  $S$  be the set of model-data pairs which are candidate matches.

$$S \subseteq M \times D \quad (1)$$

In landmark recognition, the initial pose estimate usually constrains the possible pairings between model and data segments, and  $S$  is considerably smaller than the complete space of pairs,  $M \times D$ . Line segments extracted from images can suffer from fragmentation and accretion. Therefore correspondence mappings may be many-to-many and the discrete space of possible correspondences is defined to be the powerset of  $S$ .

$$C = 2^S \quad (2)$$

Often  $S$  contains 50, 100 or more model-data pairs. Clearly the size of the correspondence space  $C$  is staggering.

### 2.2 Optimality

By defining a match error over the space  $C$  the goal is an optimal match  $c^*$ .

$$E_{\text{match}}(c^*) \leq E_{\text{match}}(c) \quad \forall c \in C \quad (3)$$

As in [Bev90]  $E_{\text{match}}$  combines two terms.

$$E_{\text{match}}(c) = E_{\text{fit}}(c) + E_{\text{om}}(c) \quad (4)$$

$E_{\text{fit}}$  is a residual squared error obtained by first fitting the model to the corresponding data.  $E_{\text{om}}$  penalizes matches which omit portions of the model.

**Weak-perspective Fitting:**  $E_{\text{fit}}(c)$  is a function of residual integrated squared perpendicular distance (ISPD) and mid-point squared distance (MPSD) between 2D model lines and corresponding data segments.

$$E_{\text{fit}}(c) = \sum_{s \in c} \frac{1}{L_d \sigma^2} (\text{ISPD}(s) + \tau \text{MPSD}(s)) \quad (5)$$

$E_{\text{fit}}(c)$  is normalized to fall in the range  $[0, 1]$  when perpendicular distance between matched segments is less than  $\sigma$ . Here  $\sigma = 7.0$  pixels is used. The term  $L_d$  is the cumulative length of the data segments. Regularization with MPSD resolves otherwise under-constrained cases. This term is given little weight:  $\tau = 10^{-4}$ .

Fitting for weak-perspective is closed form. A quadratic method of obtaining the best fit rotation, translation and scale is presented in [Bev90]. This is the principle reason why weak-perspective is computationally much faster than the full-perspective fitting described next.

**Full-perspective Fitting:**  $E_{\text{fit}}(c)$  is a function of residual 3D squared point-to-plane (SPPD) distance [Kum89, Kum90]. The points are the endpoints of the 3D model segments. The planes are defined by the two endpoints of data line segments and the focal point of the camera.

$$E_{\text{fit}}(c) = \sum_{s \in c} \frac{w}{L_d \sigma^2} (\text{SPPD}(s) + \tau \text{SPRD}(s)) \quad (6)$$

Squared point-to-ray distance, SPRD, plays a role analogous to that of the MPSD term for weak-perspective. SPPD is sensitive to the relative distance of a segment from the camera [Kum90] and the choice of world units (feet versus inches). The weight  $w$  compensates such that  $\text{SPPD}(S)$  approximates perpendicular distance in the image plane measured in pixels.

Solving for the 3D pose of the object which best fits the model to the data by equation 6 requires the use of an iterative method. The Levenberg-Marquardt method suggested by David Lowe [Low91] is used to insure convergence for difficult cases. To save computation, Kumar's algorithm has been reformulated in terms of state variables associated with each model-data pair  $s \in S$ . The sum of these state variables for all pairs  $s$  in a particular correspondence  $c$  determines the pose. This saves computation during matching by removing the need to loop over the complete set of pairs in  $c$  on each iteration of the pose algorithm.

**Hybrid Algorithm Fitting:**  $E_{\text{fit}}(c)$  is defined as in equation 5, but before it is computed the pose minimizing equation 6 is computed and the model is re-projected into the image plane.

**Omission Error:**  $E_{\text{om}}(c)$  is essentially identical for the weak-perspective, full-perspective and hybrid algorithms. It is a function of the fraction of 2D model line segments not covered by data line segments [Bev90]. A point on a model line segment is covered when a point on a data segment projects perpendicularly onto it.

$$E_{\text{om}}(c) = \sum_{m \in M} \frac{\ell m}{L_m} \left( \frac{e^{-\alpha p_m} - 1}{e^{-\alpha} - 1} \right) \quad (7)$$

$p_m$  is the fraction of model segment  $m$  not covered by data. The parameter  $\alpha$  attenuates the relative importance of smaller omissions. Here,  $\alpha = 1.021$  is used. This attenuates by 75% the error for  $p_m = 0.5$  relative to the error for  $p_m = 1.0$ . The model segment length  $\ell(m)$  is the 2D length for weak-perspective and 3D length for full-perspective and hybrid.  $L_m$  is the sum of the lengths of all the model segments.

## 2.3 Search

Local search algorithms specifically adapted to model matching [Bev90] are used to search the correspondence space  $C$  for an optimal correspondence  $c^*$ . In general, a local search algorithm moves from an initial solution, via transformations, to one that is locally optimal [Ker72, Lin73, Pap82]. It may not find the globally optimal match. However, using multiple trials of local search initiated from independently chosen random starting points the probability of missing the optimum can, in principle, be made arbitrarily small.

The notion of neighborhood is basic to local search. Here, the neighborhood consists of all correspondences which differ from the current by a single model-data pair  $s \in S$ . A match is locally optimal if it is equal to or better than all its neighbors.

For weak-perspective a *steepest-descent* strategy is used:  $E_{\text{match}}$  is computed for each neighbor and the one yielding the greatest improvement is picked. For full-perspective, where testing neighbors involves comparatively expensive 3D pose determination, a caching strategy named *inertial-descent* is used [Bev92]. All neighbors better than the current correspondence are stored in order of relative improvement. They are then sequentially tested and adopted if improvement still holds.

The hybrid-perspective algorithm is similar to the steepest-descent strategy used with weak-perspective. The neighborhood is evaluated using weak-perspective and the best neighbor using the weak-perspective  $E_{\text{match}}$  is selected to be the new match. However, when the new match becomes the current match model pose is recomputed and the model is re-projected. This can lead to cases where  $E_{\text{match}}$  is better before reprojected but worse afterwards. To prevent premature termination in such cases, a variant of *variable-depth* [Pap82] search is used. Here, if the state being left is better, then it is remembered for some fixed number of additional moves. If at any point in this exploration, a correspondence better than the one being remembered is found, the algorithm proceeds as before, otherwise it reverts to the remembered correspondence and terminates.

Figure 4a illustrates a single trial of the hybrid algorithm. The model segments are denoted by letters: the data segments by numbers. The randomly chosen initial correspondence is indicated by the first row in the table. Successive rows show improvements made by the algorithm. The match error is shown on the left. The variable-depth provision may be seen in the increase in match error between rows 1 and 2. The final solution is in fact the correct, globally optimal, match.

The local search algorithms may not find the globally optimal match  $c^*$  on a single trial. However, the probability of failure over a set of trials is conjunctive. Consequently a local search procedure with a relatively small probability of succeeding on a single trial will, over a series of sufficient length, succeed with very high probability.

Formally, let  $P_s$  be the probability of successfully finding the global optimum on a single trial. The conjunctive probability of failing to find the global optimum in  $t$  independent trials is  $Q_f$ :

$$Q_f = P_f^t \quad P_f = 1 - P_s \quad (8)$$

Therefore, the number of trials required to find the global optimum with probability  $Q_s$ , using a local search algorithm with probability of success  $P_s$ , is given by the following equation.

$$t_s = \lceil \log_{P_f} Q_f \rceil \quad Q_f = 1 - Q_s \quad (9)$$

To illustrate, if  $P_s = 0.10$  then 29 trials are required to find the optimum with probability  $Q_s = 0.95$ . In the experiments that follow,  $P_s$  is estimated for a series of landmark recognition problems based upon the usually valid assumption that the best match found in a large number of trials is the globally optimal match.

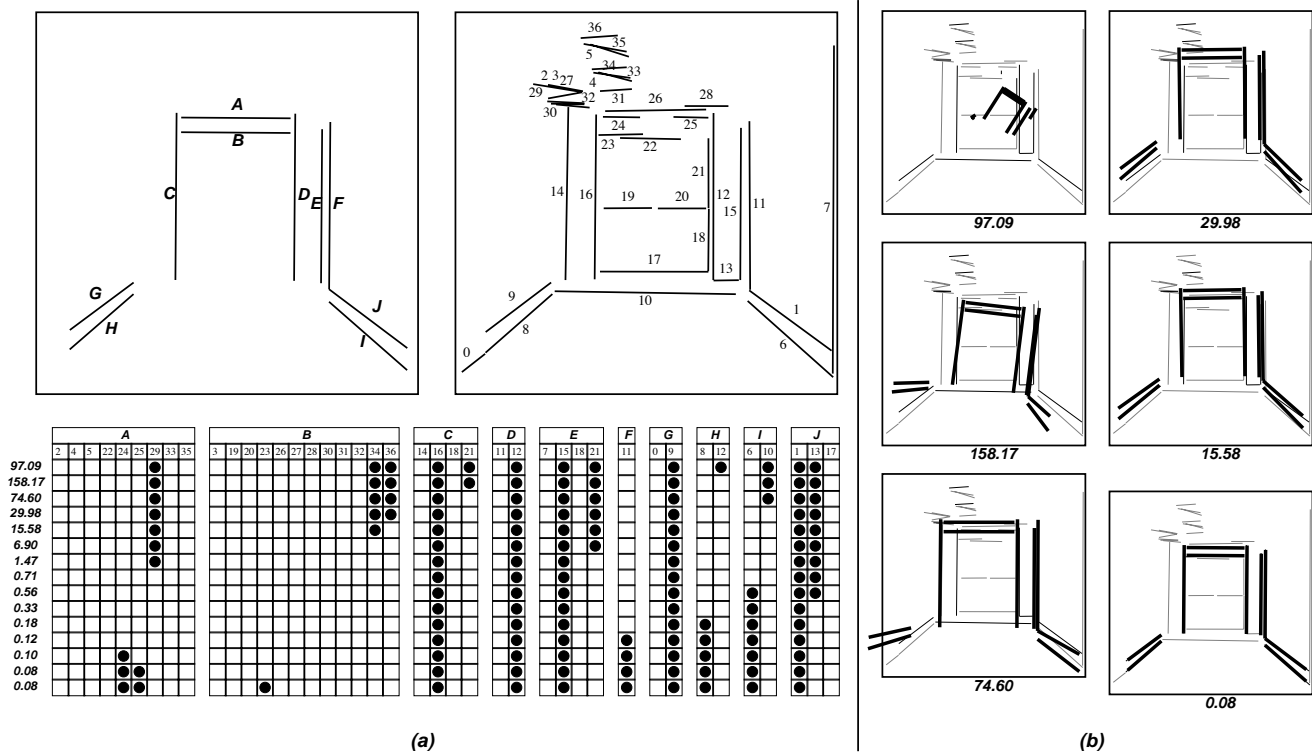


Figure 4: Landmark model, data and example matching search trace for the hybrid algorithm. (a) Model segments are designated with letters, data segments with numbers. The table indicates successively better matches found by the search algorithm. Match error is shown on the left and the final row indicates the optimal match. (b) Illustrating the reprojected landmark for intermediate matches during search and finishing off with the optimal match.

### 3 Experiments

Each of the three algorithms is tested on the task of recovering the true pose, pose 5, from each of the estimates shown in Figure 3a<sup>1</sup>. 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 forward and backward 4 feet and side-to-side 2 feet.

The predicted appearance of the landmarks, shown in Figure 3b, is used to determine the set of candidate pairs  $S$ . Candidates  $S_{\rightarrow}$  assume *directed* segments and a pair  $s = (m, d) \in M \times D$  is an element of  $S_{\rightarrow}$  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.

<sup>1</sup>In this experiment we only consider the position portion of the pose estimate associated with a match.

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

Table 1: The number of candidate pairs for each of the nine initial pose estimates and using directed,  $S_{\rightarrow}$ , and undirected,  $S_{\leftarrow}$  segments.

As a check on the importance of contrast, candidates  $S_{\leftarrow}$  are generated with the contrast constraint ignored. The size of these sets is shown in table 3. The other bounds are picked based on experience with the domain. In particular, 128 pixels is one quarter the distance across the full 512 by 512 image and is adequate to ensure the correct match is contained in the resultant search space.

Both the full-perspective and hybrid algorithms *recovered the true pose*. For a total of 18 problems, 9 poses and directed/undirected segments, both algo-

rithms reliably found exactly the same optimal correspondence, and hence recovered the robot's true pose to within the accuracy bounds of our pose algorithm [Kum90]. The weak-perspective algorithm did equally well for initial pose estimates 4, 5 and 6 where perspective has little effect. For the other cases performance was less reliable, differing from the true pose by 1 to 2 feet in 5 of the 6 cases and by nearly 8 feet in one case.

The amount of computation required to find these matches is estimated from data recorded over many trials. Between 100 and 300 trials were run in order to estimate the probability of success  $P_s$  for each of the three algorithms on each of the 18 problems. The number of trials required to find the optimal match with 95% confidence,  $t_s$ , is computed using equation 9. Multiplying  $t_s$  by the average time per trial yields the expected amount of time required to find the optimal correspondence. All times reported are in seconds running on a TI Explorer Lisp Machine.

For directed segments the results are as follows. Required trials  $t_s$  for weak-perspective on pose estimates without perspective distortion range between 7 and 10. However,  $P_s$  drops as low as 0.01 for pose 9 yielding  $t_s = 299$ . The inability of weak-perspective to correct for perspective reduces both the quality of the optimal match and the relative frequency with which it is found. The weak-perspective algorithm is fast, requiring between 0.5 and 1.0 seconds per trial.

For the full-perspective and hybrid algorithms  $P_s$  varies between 4 and 19 and between 6 and 15 respectively. The average time per trial ranges between 7.9 and 14.2 seconds and 0.8 and 1.6 seconds respectively. The hybrid algorithm is clearly the more useful, with  $P_s$  comparable to full-perspective and run times not much higher than for weak-perspective.

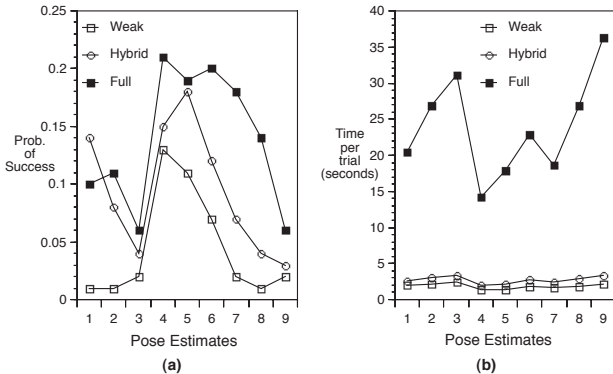


Figure 5: The estimated probability of success (a) and average run times per trial (b) for undirected segments.

For undirected segments Figure 5 shows the estimated probability of success  $P_s$  and the average time per trial broken down by pose estimate. The overall pattern is similar to that for the directed case, but with an overall decrease in  $P_s$  and increase in average run time. Very roughly, doubling the number of pairs in moving from the directed to undirected segments roughly squares the expected time required. Run times for the hybrid algorithm on the directed segments range between 6 and 20 seconds. For the undirected segments the range is between 35 and 336 seconds.

For the hybrid algorithm the estimated time required is much lower than for full-perspective, which ranges between 267 and 1783 seconds. Figure 5 shows that  $P_s$  is higher for full-perspective, suggesting that accounting for perspective when choosing between alternative moves improves the chances of finding the global optimum on a single trial. However, this benefit does not make up for the greatly increased cost of computing pose for neighboring correspondences. The average run time per trial for full-perspective is nearly an order of magnitude higher than for the hybrid algorithm.

A harder problem arises when the pose for image 1 is given as the initial estimate when the true pose is that for image 2. These two poses differ by over 10 feet in distance and several degrees in orientation. For this problem, only the full-perspective and hybrid algorithms are tested. Given sufficient trials, in the range of 100 to 300, both algorithms find optimal matches which recover true pose to within half a foot (see Figure 6). Since the initial estimates differ by more than 10 feet and several degrees in orientation these are excellent results. Moreover, realize that in one case half the expected landmark is not visible and in the other many unexpected features are visible.

For the hybrid algorithm  $P_s$  is 0.02 and 0.01. The harder problem,  $P_s = 0.01$ , is for the pose estimate of image 1 applied to the data of image 2. The times required to run  $t_s$  trials are 89 and 150 seconds respectively. The full-perspective algorithm has a higher  $P_s$ : 0.17 and 0.06 respectively. However, because it requires considerably more time per trial,  $t_s$  trials require 278 and 296 seconds respectively. All these predicted times fall under 5 minutes on a TI Explorer Lisp Machine. This is most encouraging when one considers that more efficient implementations on newer hardware might drop these times by one or perhaps even two orders of magnitude.

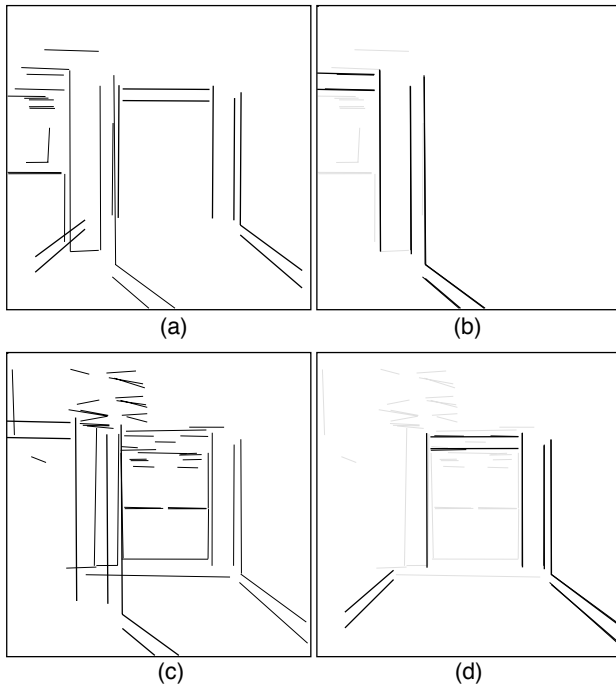


Figure 6: Confusing poses for images 1 and 2: (a) landmark projected over image 2 data as it would appear from image 1 pose, (b) successful recovery of correct match and pose, (c) landmark projected over image 1 data as it would be seen from image 2 pose. (d) successful recovery of correct match and pose.

## 4 Conclusion

Mixing weak-perspective and full-perspective in a hybrid local search strategy has produced a new algorithm for matching 3D landmark features to 2D image features. This new algorithm incorporates the best features of each: with computational requirements nearly equal to those of the comparatively inexpensive weak-perspective technique, while handling 3D perspective effects as well as the considerably more expensive full-perspective algorithm.

## Acknowledgements

We thank all those who have helped in this work, including Bob Collins, Bruce Draper, Al Hanson, Harpreet Sawhney and Rich Weiss. We especially thank Teddy Kumar, whose work on 3D pose determination has made this work possible.

## References

[Bur86] J. B. Burns, A. R. Hanson, and E. M. Rise-

man. Extracting straight lines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-8(4):425 – 456, July 1986.

[Bev92] 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.

[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.

[Fen90] Claude Fennema, Allen Hanson, Edward Riseman, J. R. Beveridge, and R. Kumar. Model-directed mobile robot navigation. *IEEE Trans. on Syst., Man, Cybern.*, 20(6):1352 – 1369, November/December 1990.

[Kum89] 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.

[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.

[Ker72] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell Systems Tech. Journal*, 49:291 – 307, 1972.

[Lin73] S. Lin and B. Kernighan. An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21:498 – 516, 1973.

[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.

[Pap82] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*, chapter Local Search, pages 454 – 480. Prentice-Hall, Englewood Cliffs, NJ, 1982.