Optimal Geometric Model Matching Under Full 3D Perspective [*†]

J. Ross Beveridge Edward. M. Riseman

Abstract

Model-based object recognition systems have rarely dealt directly with 3D perspective while matching models to images. The algorithms presented here use 3D pose recovery during matching to explicitly and quantitatively account for changes in model appearance associated with 3D perspective. These algorithms use random-start local search to find, with high probability, the globally optimal correspondence between model and image features in spaces containing over 2^{100} possible matches. Three specific algorithms are compared on robot landmark recognition problems. A full-perspective algorithm uses the 3D pose algorithm in all stages of search while two hybrid algorithms use a computationally less demanding weak-perspective procedure to rank alternative matches and updates 3D pose only when moving to a new match. These hybrids successfully solve problems involving perspective, and in less time than required by the full-perspective algorithm.

[*]This work was supported by the Defense Advanced Research Projects Agency under contract DAAE07-91-C-R035 monitored by TACOM, contract DAAH04-93-G-422 monitored by ARO, and by the National Science Foundation under grant CDA-8922572.

[†]To Appear in CVGIP:IU in 1995

Contents

1 Introduction

Algorithms which match geometric object models to image features often use alignment to test if corresponding model and image features properly match. Unfortunately, quantitative changes in model appearance due to 3D perspective are typically neglected. Consequently, these algorithms are of limited use in domains where perspective significantly alters 2D appearance. This paper presents new algorithms which account directly for changes in appearance due to perspective. These algorithms require an initial rough estimate of the object's placement relative to the camera. However, this estimate may be imprecise and may significantly alter the 2D appearance of the object as will be shown in Figures 3 and 9 below.

For several years, we have worked on an approach to matching using local search [BWR89, BWR90] which now quantitatively handles 3D perspective [BR92a, BR92b, Bev92]. A detailed account of this work may be found in Beveridge's Ph.D. dissertation [Bev93]. Matching is cast as the problem of finding the best *correspondence* between model and image features subject to a global least-squares fitting constraint. Local search algorithms developed specifically for this task use a generate-and-test strategy to explore the combinatorial space of matches.

Replacing, or supplementing, the closed-form 2D fitting techniques presented in [BWR90] with an iterative non-linear algorithm developed by Kumar [Kum89, Kum92, KH94] is the key to handling 3D perspective. 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 model relative to the camera. When a mobile robot uses this process to match known landmarks it can precisely locate itself in the world.

To the best of our knowledge, the algorithms described below are the only algorithms with a demonstrated capacity to find the *best* match in geometric matching problems involving significant perspective effects. This has been accomplished through smooth integration of effective pose recovery and combinatorial search. This ability is valuable in a number of application domains and it is needed to solve the hallway-domain landmark recognition problems presented in this paper.

2 Previous Work

Previous work on object recognition, pose determination and heuristic combinatorial optimization techniques are all pertinent to this paper. The local search techniques used here trace their roots to the work of Kernighan and Lin [Lin65, KL72, LK73]. A succinct description of local search appears in [PS82]. Kumar's pose determination algorithm [Kum89, Kum92, KH94] was the obvious choice for this work given our close collaboration on robot navigation problems [FHR+90], but in principle any 3D pose algorithm with similar inputs, outputs and performance could be used. A large body

of recognition work avoids perspective problems by using range data [Ram90]. However, this review will limit itself to prior work on optical imagery and the extent to which this work has dealt with perspective.

2.1 Four General Approaches to Recognition

The following taxonomy, while by no means complete, does help organize prior work into four broadly identifiable categories:

Key-feature: Key-feature algorithms search for highly distinctive local geometric features which indicate an object's placement.

Generalized Hough: Generalized Hough and pose clustering algorithms search in the space of object model to image pose transformations.

Tree Search: Tree search algorithms expand an interpretation tree using local geometric consistency constraints for pruning.

Geometric Hashing Geometric hashing uses local geometric features and a precompiled hash table to predict object identity and placement.

Key-feature Search An early example of the key feature approach is the 'local feature focus' method proposed by Bolles and Cain [BC82]. David Lowe [Low80, LB83, Low85] extended and further motivated this approach. A variety of key-feature algorithms have used either straight line or local curvature properties to identify object silhouettes [AF86, KJ86, GTM89, AD90]. Huttenlocher [HU90] uses feature triples to hypothesize matches under scaled orthographic projection, ranks triples according to distinctiveness, and then searches from most distinctive to least. A general problem is that overly simple key-features can become indistinctive, while richly distinctive key-features contain many parts and are consequently expensive to detect. Sitarman and Rosenfeld [SR89] have theorized that there is some 'optimum' size for key-features, but how best to select and detect key-features remains an open question.

Pose Space Search Generalized Hough transform [DY80, Bal81, Dav82] and pose clustering [Sto87] algorithms shift search from correspondence space and into pose space. Illingworth and Kittler [IK88] review generalized Hough work through 1988. Explicit representations of pose space place practical limits on the dimensionality of the pose space and most algorithms match 2D rigid objects. Notable exceptions include Silberberg [SHD86] who limited camera placement and Thompson [TM87] who used subspace projection. Stockman's [Sto87] pose clustering avoids explicit representations of pose space and a nice illustration of pose clustering can be found in the work of Hwang [Hwa89]. The noise sensitivity of generalized Hough algorithms has been assessed by Brown [Bro82] and Grimson [GH88].

Grimson concludes that false positive rates can become intolerably high in cluttered settings or when subspace projection is used.

Tree Search Early uses of tree search include tactile sensing by Gaston [GLP84] and 2D point matching by Baird [Bai85]. Grimson has extensively studied tree search [Gri90a, Gri90c, Gri90b] and shown that for 2D-rigid matching with a stopping criterion, the average-case computational complexity of tree search is $O(n^2)$, where n is the number of model features times the number of data features. Tree search is exponential without the stopping criterion. Beuler [Bre90] and Cass [Cas92] overcome some of these difficulties with *pose equivalence analysis*, which simultaneously searches in correspondence and pose space for maximal sets of pose consistent features. For scaled 2D models, the algorithm's complexity is $O(k^4m^4d^4)$ for m model features, d data features and k sided polygonal uncertainty regions about each feature. As Cass notes, while the polynomial bound is of tremendous theoretical interest, it is too high for the algorithm to be put to direct practical use.

Geometric Hashing Not unlike key-feature search, geometric hashing [KSSS86, LW88, LSW90] seeks image features which predict the presence of an object. Unlike the other three approaches discussed above, geometric hashing uses these feature to predict which of a set of objects is present. This makes geometric hashing a more general approach than the other three. However, questions remain regarding the sensitivity of geometric hashing to noise and clutter [GH90, CHS90]. Another difficulty is the need to select sets of features all belonging to the same object. Stein [SM90] does this using consecutive sides of a polygonal boundary for 2D recognition, but for 3D object models [SM92] Stein argues that hashing runs into difficulties and topological constraints between image features are more important.

2.2 Previous Work and 3D Perspective

Our work differs from most prior work in 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. Often 2D models are used, and if there is a 3D model, it is first projected into the image once to create a 2D model. These 2D techniques vary object appearance with affine transformations. Combinations of rotation, translation and scale result in a weakened form of perspective imaging called here *weak-perspective*. Fixing scale yields *2D-rigid* imaging. It is impossible for a 2D model to account for many potential changes in the pose of a 3D object. Unlimited variations in pose can be handled by mapping 3D features to the image using scaled-orthographic projection. However, this approximation does not induce vanishing points and will always distort the appearance of surfaces not parallel to the projection plane.

Table 1 lists a sampling of researchers who have used each imaging model, and suggests how few have generalized to full-perspective imaging. The complexity of computing 3D object pose under full-perspective partially explains this lack of generalization. However, more basic is the common reliance 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 [LW88], 2 points are required to establish a basis under weak-perspective while full-perspective requires 5 points.

Imaging	**Used by** ...
2D-rigid Flat objects perpendicular to the camera. The distance from the camera is fixed.	Kalvin [KSSS86] Grimson [GLP87] Stockman [Sto87] Hwang [Hwa89] Beveridge [BWR89]
Weak-perspective 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 [AF86] Gottschalk [GTM89] Costa [CHS90] Grimson [Gri90b] Beveridge [BWR90] Cass [Cas92]
Scaled-orthographic Any 3D object viewed from any arbitrary viewpoint, but only flat objects parallel to the image plane are undistorted.	Lowe [Low85] Thompson [TM87] Huttenlocher [HU90]
Full-perspective Any 3D object viewed from any arbitrary viewpoint. An initial approximate pose estimate is required.	Lowe [Low91] Beveridge [BR92b]

Table 1: Previous work by imaging model.

3 Local Search Matching

Local search refers to combinatorial optimization techniques which iteratively search a locally defined neighborhood until they arrive at locally optimal solutions [PS82]. 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.

Three specific local search matching algorithms have been developed to solve matching problems involving 3D perspective. Each uses a somewhat different neighborhood definition and somewhat different ways to choose improved matches. The *full-perspective* algorithm *always* uses Kumar's pose algorithm as it is described in Section 3.3. The *hybrid* algorithm uses the computationally more efficient weak-perspective fitting described in Section 3.2 to test neighboring matches, and recomputes 3D pose only when adopting a new match as superior to the current. The *hybrid-subset* algorithm extends the hybrid using a technique presented originally in [BWR90] for escaping local minima called subset convergent local search.

Starting search from randomly selected initial matches is necessary for these algorithms to succeed. Let P_s be the probablity of reaching the best match from a randomly chosen start in one trial. Then the probability, Q_f, of failing to see the best match in t trials decreases exponentially with the number of trials t:

$$Q_f = (1 - P_s)^t \tag{1}$$

Given an estimated probability of success $\hat{P}_s$, the estimated number of trials $\hat{t_s}$ needed to find the best match with confidence Q_s is given by the following equation.

$$\hat{t_s} = \lceil \log_{(1-\hat{P_s})} Q_f \rceil \quad \text{where} \quad Q_f = 1 - Q_s \tag{2}$$

Typically, a relatively small number of trials are needed. For example, if $\hat{P}_s = 0.4$, then 6 trials are needed to find the best match with 95% confidence. If $\hat{P}_s$ drops to 0.2, then 14 trials are needed. The key to applying local search to a new problem domain is to first estimate $\hat{P}_s$ for sample problems and then pick a number of trials sufficient to reliably solve all the example problems.

There is a tradeoff between time spent trying to get out of local optima on a single trial versus the number of trials which can be run in a fixed amount of time, and so the algorithm with the highest P_s may not be the best. As will be shown, the full-perspective algorithm finds the best match on a single trial more often than does the hybrid algorithm. However, the hybrid runs trials so much faster that it is more likely to find the best match given the same amount of time.

The analysis below shows how to empirically estimate $\hat{P}_s$ and expected run-times for alternative algorithms. This section begins by defining the combinatorial space of matches and the two match errors used: weak-perspective and full-perspective. Next the hallway test domain is introduced, followed by the three local search algorithms. Finally, their relative performance is compared on two sets

of recognition problems, one involving modest errors in initial pose and the other involving more substantial errors.

3.1 The Optimal Match Minimizes Fit plus Omission

The first step in defining a matching problem is to select a set of potentially matching object model and image features. This is done once prior to matching based upon the relative position of the model and image features when the model is projected into the image from the initial pose estimate. 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 $d \in D$.

$$C \;=\; 2^S \qquad\qquad S \subseteq M \times D \tag{3}$$

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

$$E_{\text{match}}(c^*) \;\leq\; E_{\text{match}}(c) \quad \forall c \in C \tag{4}$$

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

$$E_{\text{match}}(c) \;=\; \left(\frac{1}{\sigma^2}\right) E_{\text{fit}}(c) + E_{\text{om}}(c) \tag{5}$$

The first, E_{fit}, is a residual squared-error obtained by first fitting the model to the corresponding data. The second, E_{om}, penalizes matches which omit portions of the model from the match. The weighting coefficient σ controls the relative importance of these factors. The omission error E_{om} is defined to be a non-linear function of the fraction of the model line segments not covered by data line segments [BWR90]. Omission is measured in the image plane with the model transformed to the best-fit configuration relative to the data. A point on a model line segment is covered when a point on a data segment projects perpendicularly onto it. Unfortunately, space here does not permit a complete accounting of match error parameterization. The reader is referred to Beveridge's Ph.D. dissertation [Bev93] for this level of detail.

3.2 Fitting 2D Projections of 3D Models

A 2D projection of a 3D model may be fit by minimizing 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. A weak-perspective match error may be defined as

$$E_{\text{match-wp}}(c) \;=\; \left(\frac{1}{\sigma^2}\right) E_{\text{fit-wp}}(c) + E_{\text{om}}(c) \tag{6}$$

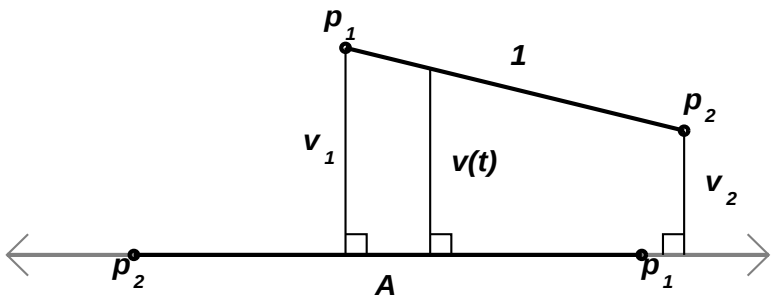

Figure 1: Points on data segment 1 project perpendicularly onto model line A. The perpendicular distance at any point along 1 may be written as a function $v(t)$.

where $E_{\text{fit-wp}}(c)$ is a function of the residual ISPD.

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:

$$v(t) \;=\; v_1 + (v_2 - v_1)\,\frac{t}{\ell} \qquad\qquad 0 \le t \le \ell. \tag{7}$$

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

$$\text{ISPD} \;=\; \int_0^{\ell} v^2\,(t)\,dt \;=\; \frac{\ell}{3}\left(v_1^2 + v_1 v_2 + v_2^2\right). \tag{8}$$

It is possible to derive a closed-form solution for the rotation, translation and scale which minimizes the sum of ISPD for essentially any model matched to any arbitrary set of image line segments. The resulting equation is second order [BWR90] with coefficients defined in terms of the endpoints of model and image line segments. Interestingly, the case of fixed scale fitting is still closed-form but quartic rather than quadratic [BWR89]. These fitting equations are fully developed in Beveridge's Ph.D. dissertation [Bev93] along with two additional improvements: 1) a regularization term which extends the number of cases for which there exists a unique best-fit solution, and 2) a state-vector formulation which facilitates efficient fitting of matches differing slightly from a current match.

3.3 Fitting Under 3D Perspective: Pose Determination

Kumar [KH89] observes 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 3D transformation between camera and model 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. A full-perspective match error may be defined as

$$E_{\text{match-fp}}(c) = \left(\frac{1}{\sigma^2}\right) E_{\text{fit-fp}}(c) + E_{\text{om}}(c) \tag{9}$$

where $E_{\text{fit-fp}}(c)$ is a function of the residual 3D point-to-plane distances used in Kumar's algorithm.

Here, three enhancements to Kumar's algorithm are utilized. First, a midpoint-to-midpoint regularizing term is added to Kumar's original measure. This added measure will yield a unique best-fit pose in many cases where such a unique pose would not otherwise be defined. The second is the use of the Levenberg-Marquardt method within the inner loop of the iterative pose algorithm. Levenberg-Marquardt [PFTV88] is a good way of solving systems of equation which may be nearly singular, and this is important because regularization makes otherwise singular cases near singular. The third enhancement is to completely rederive the 3D pose update equations in order to characterize pose in terms of a sum of state vectors: one vector per pair of corresponding line segments. This expedites the testing of alternative matches during local search. These three enhancements are more fully described in Beveridge's Ph.D. dissertation [Bev93].

3.4 Hallway Navigation – A Test Domain

Problems requiring that perspective be taken into account arise in the context of landmark-based robot navigation. 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. These features are 3D segments from a partial wire frame model of the hallway. To give a sense of scale to this illustration, pose 5 is 40 feet from the doorway at the end of the hallway. This domain is 'perspective-sensitive' because small changes in robot position introduce significant perspective effects. Pose 2 is only 2 feet to the left of pose 5, yet observe how the relative orientation of the baseboards changes between the two.

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 3.8 is to recover the true pose that produced the image of Figure 4a; each of the 9 pose estimates shown in

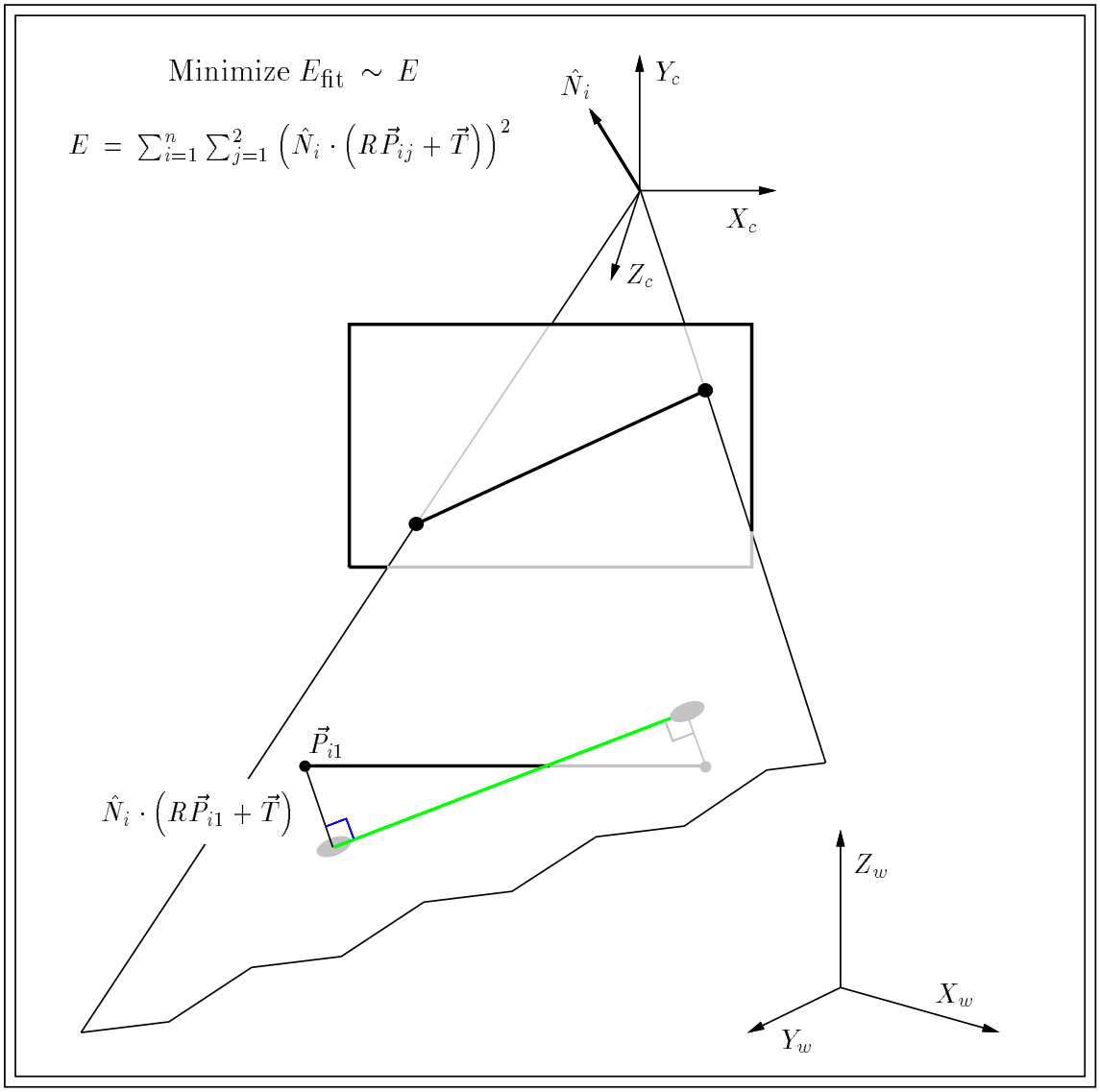

Figure 2: Point-to-plane fit measure for 3D pose algorithm. $\hat{N}_i$ is the normal to the plane defined by the ith image line. R and $\vec{T}$ are the rigid transformation from model to camera coordinates.

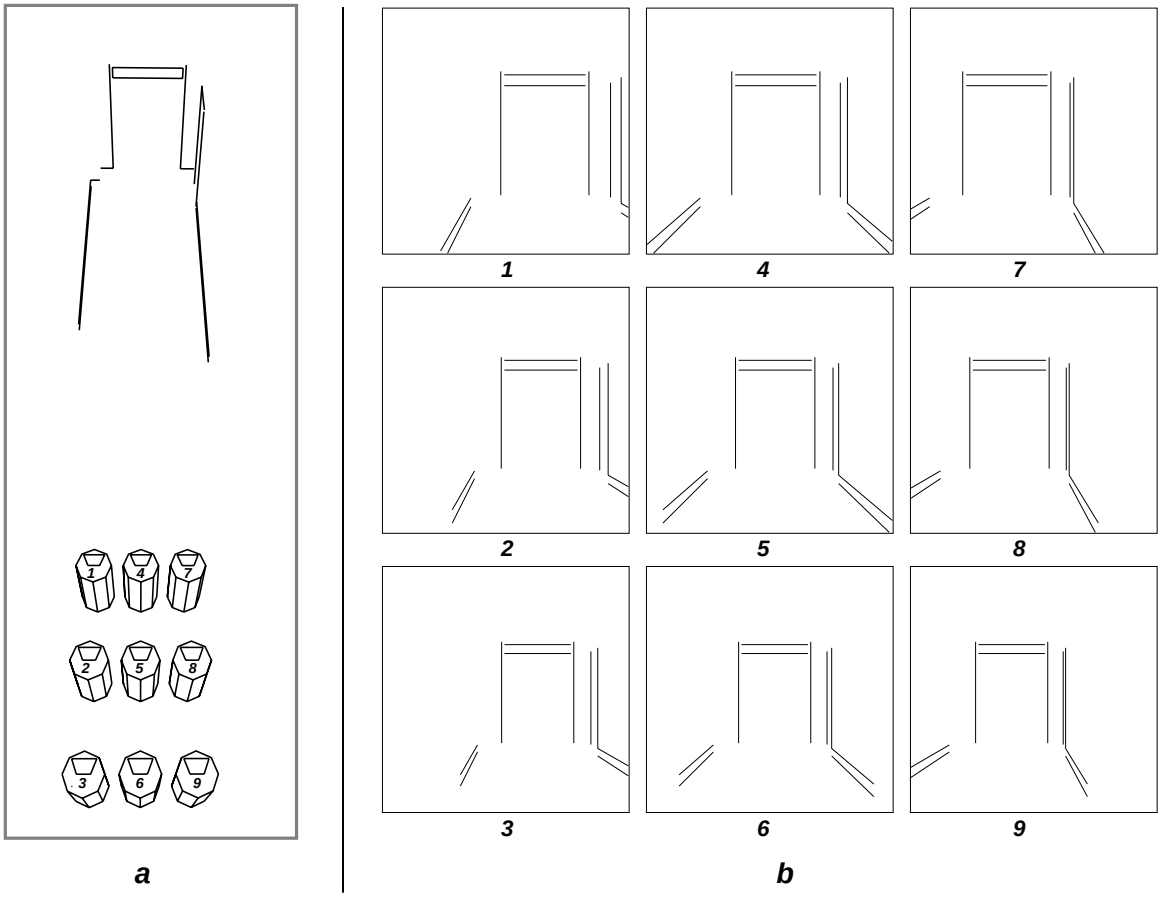

Figure 3: Perspective views of landmarks: a) Robot in relation to a partial model of the hallway. b) Partial model as it appears from each of the 9 poses. The relative orientation of the baseboards and the door frame change as the robot's pose estimate shifts laterally.

(a)

(b)

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

Figure 3 will be used as initial estimates of robot position. The experiment presented in Section 3.9 treats a harder problem; that of recovering from a larger error in initial pose estimate. The robot poses from which the two images in Figure 4 are taken differ by over 10 feet. The goal of the experiment in Section 3.9 is to recover the pose from which each of the two images in Figure 4 were taken, given the other pose as an intiial estimate.

3.5 Full-perspective Matching

Key to defining a local search algorithm is the local neighborhood definition. A particularly simple neighborhood contains all correspondences obtained by adding or removing a single pair $s \in S$ from the current correspondence c. If there are n pairs in S, then c may be represented as a bit string of length n. Using this bitstring representation, adding or removing a single pair generates a string Hamming distance 1 from the current match. The is the neighbhorhood used by the full-perspective matching algorithm.

A steepest-descent local search strategy computes $E_{\text{match}}(c')$ for all matches c' in the neighborhood of c, and provided there exists a c' with lower error, moves to the neighbor yeilding the greatest reduction in E_{match}. The full-perspective algorithm minimizes the full-perspective match error from equation 9; were it to use the steepest-descent strategy it would compute n 3D poses before every move. To save computation, a modified strategy named here *inertial-descent* is used. 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. The algorithm applies the first transformation and moves to a new, and guaranteed to be improved, match state. Then, rather than completely evaluate the n new neighbors of this new match, the algorithm tests whether the second ranked transformation leads to a better match. If it does, it moves and repeats this process with the third ranked transformation, and so on. This is done until either the list of transformations is exhausted or a transformation is encountered which no longer yields improvement. Once this occurs, then all n neighbors are tested and a new list built. The algorithm terminates when none of the n neighbors are better than the current match.

An example of the full-perspective algorithm is presented below using Figures 5 and 6 for illustration. The matching problem is to match the landmark model shown in Figure 3 to data extracted from Image 2, shown in Figure 4b. The initial pose estimate is pose 5, shown in relation to the 3D landmark model in Figure 3a. Labeled model and data line segments are shown on the right hand side of Figure 5. The model line segments, labeled with letters, are shown projected into the image plane as they would appear from pose 5. The data line segments are extracted from the image using the Burns algorithm [BHR86].

Figure 5 also shows a complete trace of two independent runs of the inertial-descent algorithm.

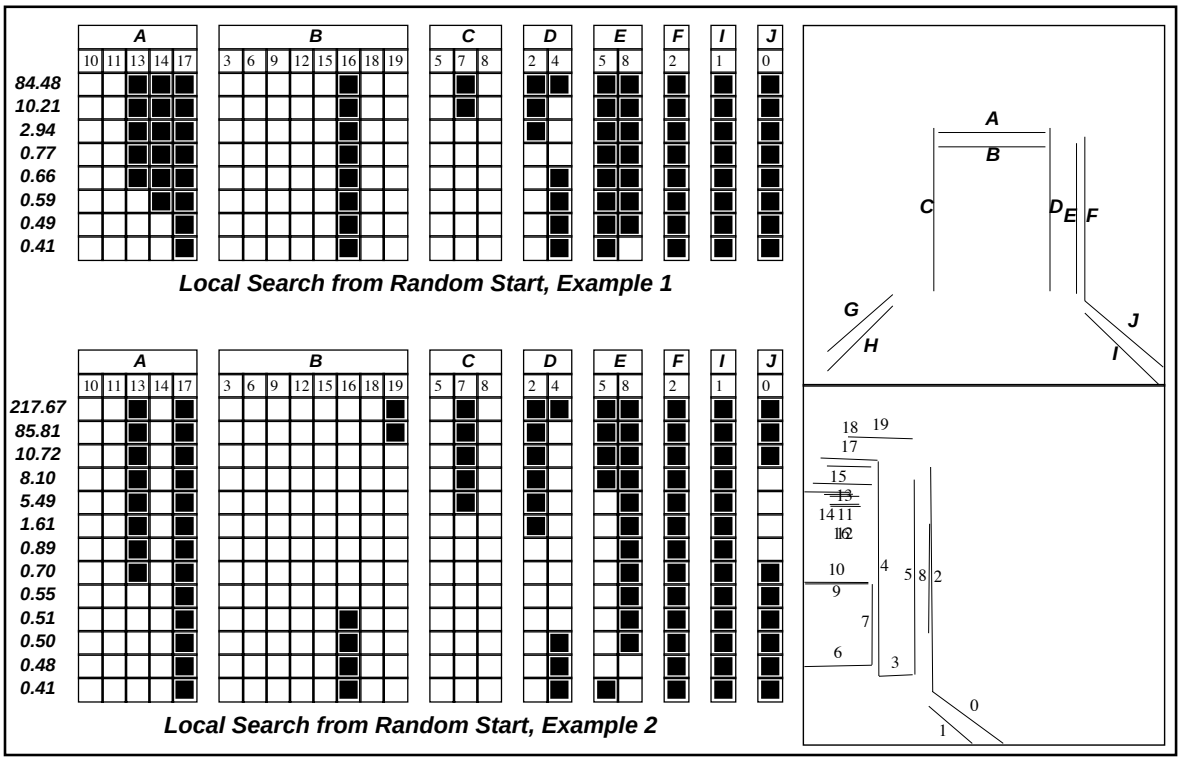

Figure 5: Example of the full-perspective algorithm.

Table 2: The improvement lists generated by the inertial-descent algorithm each time it tests adding/removing a single pair. The highlighted pairs, applied in sequence, lead to improved matches.

Match	Improvements in Descending Order
84.48	$((\mathbf{D},\mathbf{4})(D,2)(E,5)(I,1)(J,0)(B,15)(B,12)\ldots$
10.21	$((\mathbf{C},\mathbf{7})(J,0)(I,1)(E,5)(B,12)(B,15)(A,11))$
2.94	$((\mathbf{D},\mathbf{2})(E,5)(E,8)(A,11)(B,12)(B,15))$
0.77	$((\mathbf{D},\mathbf{4})(\mathbf{A},\mathbf{13})(\mathbf{A},\mathbf{14})(\mathbf{E},\mathbf{8})(A,17)(B,15)(B,16))$
0.41	$(\,)$

Each row indicates a successively better match derived by locally improving an initial randomly selected match indicated in the first row. The columns indicate specific candidates pairs of model-image segments from the set S. A filled in square indicates the pair is included in the current match. While both the trials shown lead to the globally optimal match, this is of course not always the case. In 100 trials, this globally optimal match was found 18 times. The maximum likelihood estimate for the probability of success on this problem, $\hat{P}_s$, is therefore 0.18. Equation 2 indicates that 15 trials are required to find this match with 95% confidence.

The combinatorial search space of possible matches, defined by the set of candidate pairs S, is initially chosen based upon proximity and compatible orientation between the model and image segments. In this example there are 23 pairs in S, of which 7 belong in the globally optimal match. The determination of which pairs of segments belong to S is made based upon the initial projection of the model into the image. Observe that based upon the intial pose estimate, model line segments G and H have no corresponding data segments with which they can match. The process of selecting candidate pairs is explained more fully in Section 3.8.

Table 2 illustrates for example 1 (Figure 5) the sorted lists of pairs whose bits, if toggled, yield improvement. The highlighted pairs at the head of the lists actually lead to improved matches. For the first three matches, inertial-descent didn't save any computation relative to steepest-descent. From the new match obtained by toggling the first pair on the list, the second pair no longer improved the match, and therefore all n neighbors of the new match were tested. However, for the fourth match, the list allowed the algorithm to apply four changes in correspondence in succession without expending effort testing alternatives. This reduced by nearly 50% the number of 3D pose calculations required to solve this example.

To emphasize that new 3D poses are generated during search, Figure 6 shows the projection of the landmark from these updated poses for the successively better matches found in example 1, Figure 5. The initial projection looks genuinely awful, as is to be expected given an initial random assignment of landmark to image features. It is the view of the landmark as it would appear to a robot downstairs and in a room off to the right of the hallway: presuming the robot could see through floors and walls.

Showing this initial projection emphasizes that it need not itself be good, or even logical. What matters only is that local search is able to discover a path from this initial match to one which is better. This is the case here, and by the fourth match the match error has dropped from 84.48 to 0.77 and the projection is looking reasonable. The last four steps in the search path do not dramatically alter the appearance of the projected landmark. However, they do represent refinements in the correspondence mapping which improve the accuracy of the final robot pose estimate.

Explaining why a particular trial of local search does or does not move to the best match is difficult. The fit error, omission error and current pose estimate interact in complex ways. As the

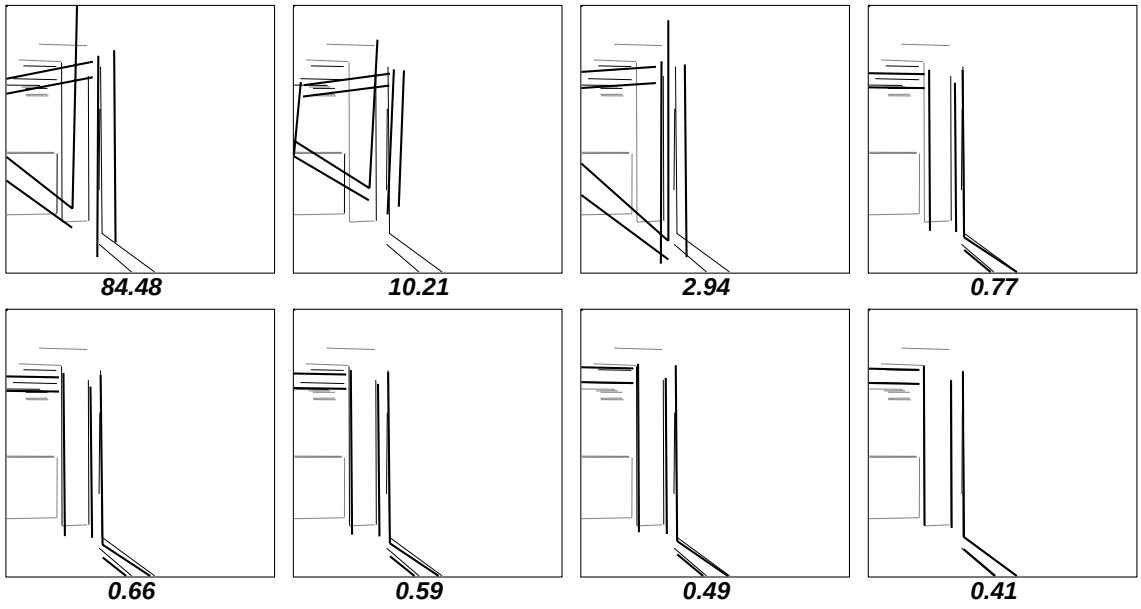

Figure 6: Landmark projections for full-perspective example.

traces in Figure 6 show, pairs are dropped from the match, added to the match, and sometimes the same pair is first dropped and later added. A common pattern is the initial move drop pairs most incompatible with the current match and later moves both add and drop pairs. While explaining or predicting what will happen on a single trial is difficult, it is also not that important. What matters is that reliable estimates of $\hat{P}_s$ be obtained for a domain, and these estimates can typically be derived from empirical tests on representative training problems.

3.6 Hybrid Matching

Even using inertial-descent rather than steepest-descent, the cost of computing the 3D pose for all n neighbors makes the full-perspective algorithm rather slow. In rough terms, it is an order of magnitude faster to fit the projection of the landmark using the closed-form weak-perspective fitting algorithm than it is to compute the 3D pose using Kumar's algorithm. The hybrid algorithm uses the cheaper fitting algorithm to rank neighboring matches and the 3D pose algorithm only when a new match is adopted. Since a single trial of steepest-descent local search tests kn matches, where n is the neighborhood size and k the number of moves from the initial random match to the final local optima, the hybrid saves a great deal of computation. Using typical numbers from the experiments which follow, for $n = 100$ and $k = 25$, a potential $2,500$ 3D pose updates is reduced to 25.

The hybrid algorithm minimizes the weak-perspective $E_{\text{match-wp}}$ from equation 6 and updates the 3D pose and hence 2D model appearance after each move. Therefore, $E_{\text{match-wp}}$ will change due to the 3D pose update and may in some cases get worse. The introduces a minor added complexity.

One option is to immediately terminate search. Another is to go back and start trying other neighbors in the hopes of finding one for which $E_{\text{match-wp}}$ stays down *after* 3D pose is updated. The third alternative, and the one used here, is to remember the match that was best and continue searching from the new, worse, match.

The strategy of going on to explore matches worse than already seen is in principle somewhat antithetical to local search, which in concept *only* moves to better states. However, it is not uncommon in practice. For the hybrid algorithm, the strategy for exploring worse matches is to continue search for some fixed number of moves. In all the experiments presented here, up to 10 exploratory moves are permitted. If, during this exploration, a match better than that being remembered is found, then local search continues and the previously best match is forgotten. If nothing better is found, then the best match being remembered is returned. Checks are added to prevent cycling back to the already discovered best match, since cycling defeats the purpose of continued search. There is a variant of local search called *tabu search* [Glo89] (Glover uses spelling 'tabu') which embodies this idea: making moves leading backwards taboo.

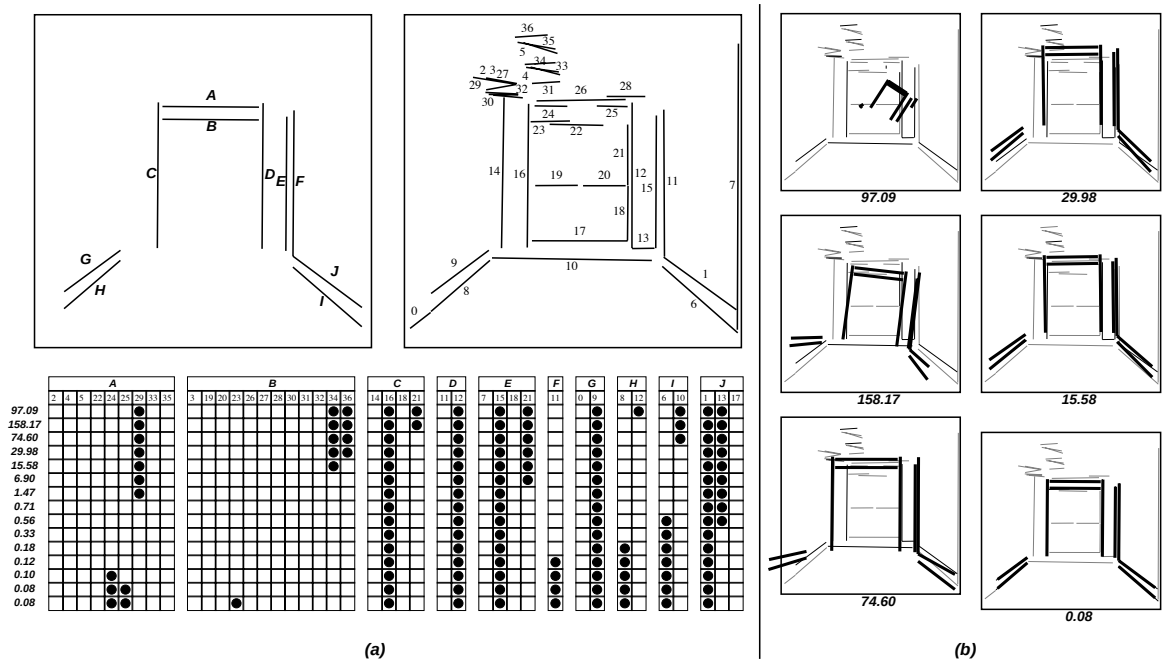

Figure 7: Example search trace for the hybrid search. a) The match error does increase in the first move, but then decreases in all successive moves. b) Landmark projections for the hybrid search. The first 5 matches projected into the image along with the final optimal match.

Figure 7 illustrates a single trial of the hybrid algorithm. Row one in Figure 7a shows the randomly chosen initial correspondence. The provision for adopting worse matches is seen in the increased match error between rows 1 and 2. The final solution in the last row is the correct, globally optimal, match. The landmark projected over the image segments for the first 5 matches and the optimal match are

shown in Figure 7b. Probabilities of success estimates $\hat{P}_s$ for problems similar to this presented below range from 0.20 to 0.45.

3.7 Hybrid Subset-Convergent Matching

The hybrid algorithm is more prone to getting stuck on local optima than is the full-perspective algorithm. To overcome this problem, the hybrid-subset algorithm initiates search for better matches from *subsets* of matches found to be locally optimal by the hybrid algorithm. 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. For a truly good match, a Hamming-distance-1 local search initiated from subsets of the match should converge back to the same match. On the other hand, if the match is globally poor, then subsets of the match are probably incompatible. The subset-convergent algorithm has been shown in some cases to dramatically increase the probability of finding the optimal match on a single trial [BWR90, Bev93].

In keeping with all our past work, search is initiated from only *four* preselected subsets. These subsets are selected using an automated heuristic procedure which selects pairs of model segments 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 the subsets used by the hybrid-subset algorithm. A fuller description of the subset-convergent control strategy for local search appears in Beveridge's Ph.D. dissertation [Bev93].

3.8 Experiment 1: Recovering from Modest Pose Errors.

Each of the algorithms introduced above is tested on the task of recovering the true pose of 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. The space of candidate pairs S for each matching problem follows from the initial placement of the model segments as shown in Figure 3b. A 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. (The image is 512x512 pixels)

3) d is at least 1/4 the length of m.

4) d and m have the same sign-of-contrast.

These tests filter candidate pairs based upon relative orientation, rough proximity, relative size and gradient direction. The thresholds are picked based upon experience with the domain and insure the correct match is contained in the resultant search space.

Table 3: Size of candidate pair sets with/without the sign-of-contrast.

	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

The sign-of-contrast filter is appropriate when segments result from surface markings rather than occlusion. For example, the top edge of the baseboards in the hallway are white on top and black on the bottom, and image segments with the opposite contrast need not be considered. Sign-of-contrast is not always a realistic filter. To test its importance, all experiments are conducted with and without the filter generating sets of pairs S and $\bar{S}$ respectively. Ignoring sign-of-contrast roughly doubles the number of pairs as shown in Table 3.

In all 18 matching problems resulting from 9 initial pose estimates and 2 sets of candidate pairs, all three matching algorithms reliably found exactly the same optimal correspondence. Consequently, the robot's true pose is recovered to within the accuracy bounds of our pose algorithm [KH90] [1].

As a point of comparison, a steepest-descent algorithm using only the weak-perspective fitting procedure was tested on this problem, and did equally well for initial pose estimates 4, 5 and 6, where the landmark projections could successfully be rotated, translated and scaled to fit the image data. However, for the other cases the 3D pose estimates derived from the best matches differed from the true pose by as much as 8 feet, confirming weak-perspective alone is inadequate for these problems.

Probability of success estimates $\hat{P}_s$ for each algorithm on each problem were calculated based upon between 100 and 300 trials. These estimates using candidate pairs filtered on sign-of-contrast are plotted in Figure 8a. Corresponding estimates without using sign-of-contrast are plotted in Figure 8b. The $\hat{P}_s$ values drop when sign-of-contrast is ignored. For instance, $\hat{P}_s$ for the full-perspective algorithm ranges between 0.17 and 0.55 with sign-of-contrast and without drops to between 0.06 and 0.22. However, if there is a surprise in these results, it is that the drop is not more precipitous. Neglecting sign-of-contrast grows the search spaces astronomically, from about 2^{50} to 2^{100} states. That local search still finds the best match reliably in 100 trials shows the strength of the approach.

Using the criterion that higher $\hat{P}_s$ is better, the hybrid-subset algorithm does better than the full-perspective algorithm on 13 out of 18 problems. It does better than the hybrid on all problems, but this is expected, since the local optima of the hybrid algorithm are a superset of those for the hybrid-subset algorithm. However, as the run-times reported next show, this apparent superiority is

[1] Only the position portion of the pose estimate associated with a match is considered. For these problems the 3D pose algorithm appears accurate to within roughly 6 inches.

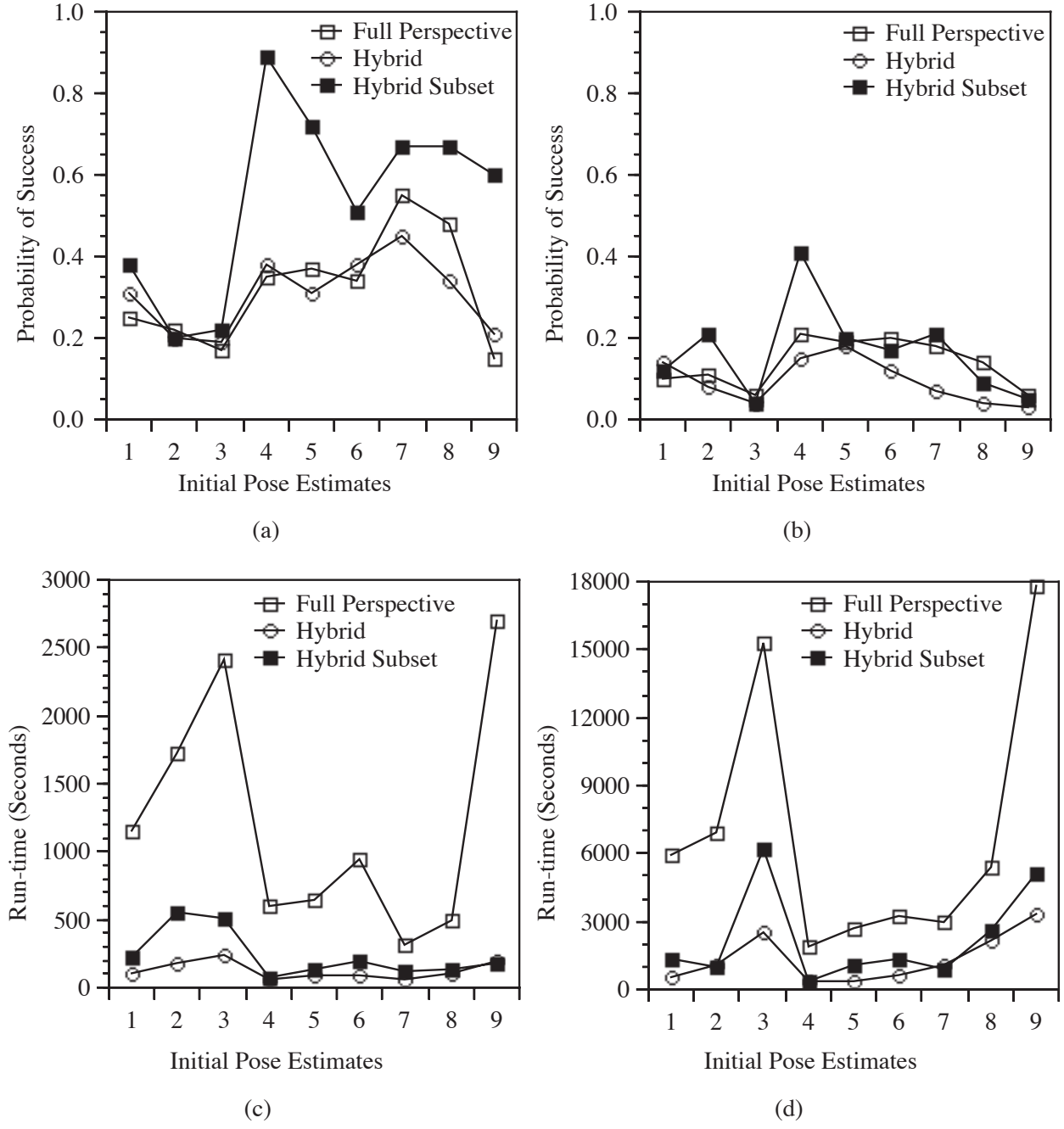

Figure 8: $\hat{P}_s$ and estimated run-time for 9 pose estimates. a) $\hat{P}_s$ with sign-of-contrast filter, b) $\hat{P}_s$ without sign-of-contrast filter, c) $\hat{r}_s$ with sign-of-contrast filter, and d) $\hat{r}_s$ without sign-of-contrast filter. Run-times for TI Explorer II Lisp Machine. A Sparc 10 is over 50 times faster.

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 in seconds on a Lisp Machine.

	Image 2 to Image 1				Image 1 to Image 2			
Algorithm.	$\hat{P}_s$	r	$\hat{t}_s$	$\hat{r}_s$	$\hat{P}_s$	r	$\hat{t}_s$	$\hat{r}_s$
full-perspective	0.29	172	9	1,548	0.18	63	18	1,134
hybrid	0.02	6	149	894	0.01	5	299	1,495
hybrid-subset	0.09	25	32	800	0.06	11	49	539

lost when absolute time to solve the problem is considered.

The estimated or expected run-time $\hat{r}_s$ needed to solve a given problem with 95% confidence is the product of the number of trials needed $\hat{t}_s$ (equation 2) and the average run-time per trial. These times are plotted for the problems with and without sign-of-contrast in Figure 8c and Figure 8d respectively. The average run-time per trial for the hybrid algorithm is sufficiently lower than for the hybrid-subset algorithm that the hybrid is faster on 14 out of the 18 problems. This makes the hybrid the algorithm of choice for these problems.

The times shown are for a T. I. Explorer II Lisp Machine, and the same algorithm in C on a Sparc 10 is over 50 times faster. Using the sign-of-constrast, the newer system can solve all these problems in under 5 seconds, and even without sign-of-contrast, just over 1 minute is required for the hardest problem.

3.9 Experiment 2: Recovering from 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.

Several factors make these matching problems difficult. First, in one case half the expected segments are not visible while in the other many unexpected segments are visible. To succeed matching must both disregard clutter and perform partial matching. Second, to account for the greater uncertainty in pose, the proximity constraint used to select the set of candidate pairs S is relaxed to include all segments d within 256 pixels of a model segment m. This increases the number of pairs to be considered and the size of the search space.

Table 4 reports $\hat{P}_s$, the average run-time per trial in seconds r, the number of trials $\hat{t}_s$ required to find the optimal match with 95% confidence, and finally the estimated run-time $\hat{r}_s$ required to confidently find the optimal match on the Lisp Machine. Comparing the run-times $\hat{r}_s$ using each algorithm, the hybrid-subset is superior to the other two. The highest of the two run-times for the

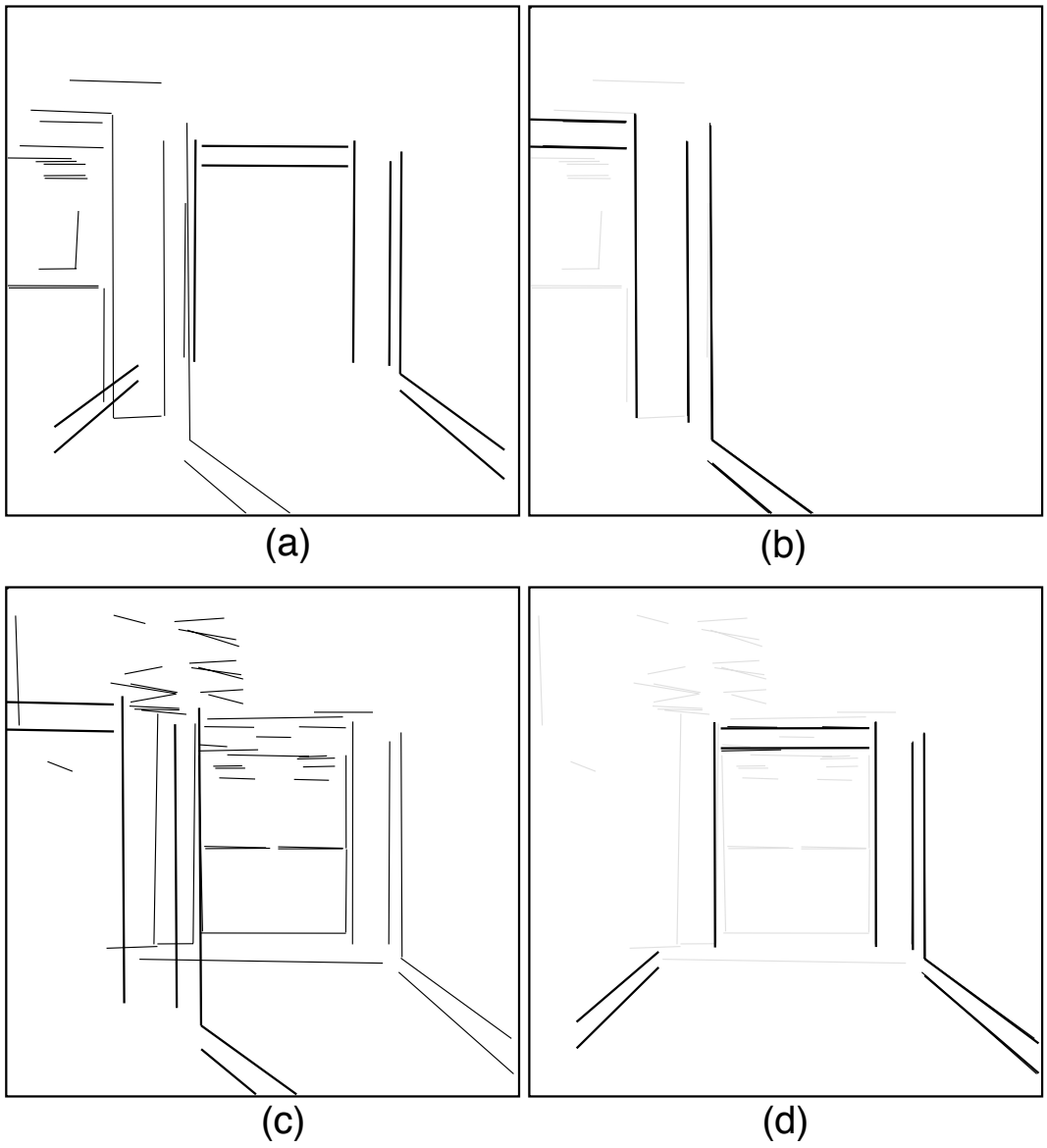

Figure 9: Confusing poses for images 1 and 2. a) landmark projected over image 2 data as it would appear from image 1 pose, b) successful recovery of correct match and pose, c) landmark projected over image 1 data as it would be seen from image 2 pose. d) successful recovery of correct match and pose.

hybrid-subset algorithm is still lower than the best time of either of the other two algorithms. Using the C version of the hybrid-subset algorithm on a Decstation 5000, $\hat{r}_s$ would drop to 16 and 11 seconds respectively.

The hybrid algorithm performed better on the modest pose errors in the previous section while the hybrid-subset algorithm performed best on the larger pose error problems in this section. This is a rather small amount of data upon which to draw general conclusions; however, it does suggest a possible trend.

4 Conclusion

The indoor robot navigation problems presented motivated us to develop the quantitatively accurate full-perspective matching algorithms presented in this paper. These algorithms are needed, in part, because those features which most accurately determine robot pose are most sensitive to perspective effects. For instance, the baseboards in the hallway constrain the left-to-right placement of the robot. Finding the doorway at the end of the hallway constrains the distance of the robot from the doorway, but offers little side-to-side constraint. Our experience suggests it simply isn't possible to reliably solve these hallway matching problems and recover robot pose using weak-perspective matching.

The systematic evaluation of alternative search heuristics presented in this paper exemplifies how local search matching can be tested in a particular application domain. These tests revealed that while introducing 3D pose determination into the matching process is essential, less computationally demanding approximations can still play a vital role. The full-perspective algorithm using 3D pose determination in all phases of search solved all the problems presented. However, the hybrid algorithms, with their sparing use of 3D pose and primary reliance upon closed-form weak-perspective fitting, solved all the same problems and in less time.

While robot navigation is one domain where perspective plays an important role, there are others. For example, augmented reality systems must determine the pose of objects relative to the observer. Future work is expected to transfer the techniques demonstrated in this paper to this and other other application domains. Current work [SB94] is also beginning to extend these techniques to mutlisensor recognition problems where object and sensor geometry simultaneously constrain object pose and sensor registration.

References

[AD90] Nirwan Ansari and Edward J. Delp. Partial shape recognition: A landmark-based approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(5):470 – 483, May 1990.

[AF86] 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.

[Bai85] Henry S. Baird. *Model-Based Image Matching Using Location*. MIT Press, Cambridge, MA, 1985.

[Bal81] Dana H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111 – 122, 1981.

[BC82] R. C. Bolles and R. A. Cain. Recognizing and locating partially visible objects: the local-feature-focus method. *International Journal of Robotics Research*, 1(3):57 – 82, 1982.

[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 Obejct Recognition: Optimal Correspondence and Pose*. PhD thesis, University of Massachuesetts at Amherst, May 1993.

[BHR86] 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.

[BR92a] 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.

[BR92b] 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.

[Bre90] Thomas M. Breuel. An efficient correspondence based algorithm for 2D and 2D model based recognition. A.I. Memo 1259, MIT, MIT AI Lab, October 1990.

[Bro82] Christopher M. Brown. Bias and noise in the Hough transform 1: Theory. Technical Report TR - 105, Computer Science Department, University of Rochester, June 1982.

[BWR89] 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).

[BWR90] 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.

[Cas92] Todd A. Cass. Polynomial-time object recognition in the pressence of clutter, occlusion, and uncertainty. In *Proceedings: Image Understanding Workshop*, pages 693 – 704, San Mateo, CA, January 1992. DARPA, Morgan Kaufman.

[CHS90] 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.

[Dav82] Larry S. Davis. Hierarchical generalized Hough transforms and line-segment based generalized Hough transforms. *Pattern Recognition*, 15(4):277 – 285, 1982.

[DY80] Larry S. Davis and S. Yam. A generalized Hough-like transformation for shape recognition. Technical Report TR–134, University of Texas, Computer Science, 1980.

[FHR+90] 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.

[GH88] W. E. L. Grimson and D. P. Huttenlocher. On the Sensitivity of the Hough Tranform for Object Recognition. In *Proc. of the International Conference on Computer Vision*, pages 700 – 706, 1988.

[GH90] W. Eric L. Grimson and Daniel P. Huttenlocher. On the Sensitivity of Geometric Hashing. In *Proccedings: ICCV 3*, pages 334 – 338, Osaka Japan, December 1990. IEEE Computer Society, IEEE Computer Society Press.

[Glo89] F. Glover. Tabu search – part i. *ORSA Journal on Computing*, 1(3):190 – 206, 1989.

[GLP84] P. C. Gaston and T. Lozano-Pérez. Tactile recognition and localization using object models: The case of polyhedra on a plane. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI – 6:721 – 741, May 1984.

[GLP87] 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.

[Gri90a] W. E. L. Grimson. The Combinatorics of Object Recognition in Cluterred Environments Using Constrained Search. *Artificial Intelligence*, 44(1):121 – 165, July 1990.

[Gri90b] W. Eric L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* MIT Press, Cambridge, MA, 1990.

[Gri90c] W. Eric L. Grimson. The Effect of Indexing on the Complexity of Object Recognition. In *Third International Conference on Computer Vision*, pages 644 – 651. IEEE, IEEE Computer Society Press, December 1990.

[GTM89] 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.

[HU90] Daniel P. Huttenlocher and Shimon Ullman. Recognizing solid objects by alignment with an image. *Internation 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.

[IK88] J. Illingworth and J. Kittler. A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 44:87 – 116, 1988.

[KH89] 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.

[KH90] 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.

[KH94] Rakesh Kumar and Allen R. Hanson. Robust Methods for Estimating Pose and a Sensitivity Analysis. *CVGIP:Image Understanding*, 11:(to appear in November), 1994.

[KJ86] T. F. Knoll and R. C. Jain. Recognizing partially visible objects using feature indexed hypotheses. *IEEE Journal of Robotics and Automation*, 2:3 – 13, 1986.

[KL72] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell Systems Tech. Journal*, 49:291 – 307, 1972.

[KSSS86] 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.

[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.

[Kum92] Rakesh Kumar. *Model Dependent Inference of 3D Information From a Sequence of 2D Images.* PhD thesis, University of Massachusetts, Amherst, February 1992.

[LB83] David G. Lowe and T. O. Binford. The Perceptual Organization of Visual Images: Segmentation as a Basis for Recognition. In *Proc. Image Understanding Workshop, Stanford*, pages 203 – 209, June 1983.

[Lin65] S. Lin. Computer solutions of the traveling salesman problem. *Bell Syst. Comput. J.*, 44:2245 – 2269, 1965.

[LK73] S. Lin and B. Kernighan. An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21:498 – 516, 1973.

[Low80] David G. Lowe. Solving for the Parameters of Object Models from Image Descriptions. In *Proc. ARPA Image Understading Workshop*, pages 121 – 127, 1980.

[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.

[LSW90] Yehezkel Lamdan, Jacob T. Schwartz, and Haim J. Wolfson. Affine invariant model-based object recognition. *IEEE Transactions on Robotics and Automation*, 6(5):578 – 589, October 1990.

[LW88] 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.

[PFTV88] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, Cambridge, 1988.

[PS82] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*, chapter Local Search, pages 454 – 480. Prentice–Hall, Englewood Cliffs, NJ, 1982.

[Ram90] Ramesh C. Jain and Anil K. Jain (editors). *Analysis and Iterpretation of Range Images*. Springer-Verlag, New York, 1990.

[SB94] Anthony N. A. Schwickerath and J. Ross Beveridge. Model to Multisensor Coregistration with Eight Degrees of Freedom. In *Proceedings: Image Understanding Workshop*, pages 481 – 490, Los Altos, CA, November 1994. ARPA, Morgan Kaufmann.

[SHD86] T. M. Silberberg, D. Harwood, and L. S. Davis. Object recognition using oriented model points. *Computer Vision, Graphics, and Image Processing*, 35:47 – 71, 1986.

[SM90] Fridtjof Stein and Gérard Medioni. Graycode representation and indexing: Fast two dimensional object recognition. In *Proc of 10th ICPR*, page ?, Atlantic City, June 1990.

[SM92] Fridtjof Stein and Gérard Medioni. Recognition of 3-d objects from 2-d groupings. In *Proceedings: Image Understanding Workshop*, pages 667 – 674, San Mateo, January 1992. DARPA, Morgan Kaufmann.

[SR89] R. Sitaraman and A. Rosenfeld. Probabilistic analysis of two stage matching. *Pattern Recognition*, 22(3):331 – 343, 1989.

[Sto87] George Stockman. Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, 40:361 – 387, 1987.

[TM87] 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.