Optimal Geometric Model Matching Under Full 3D Perspective

J. Ross Beveridge   Edward. M. Riseman

Abstract

Prior work on model-based or model-directed object recognition generally has not quantitatively dealt with 3D perspective when matching models to images, and applicability of the associated techniques is correspondingly limited. In contrast, algorithms presented here find optimal matches between 3D geometric object models and 2D image features which account explicitly and quantitatively for changes in model appearance associated with 3D perspective. This is accomplished by using a 3D pose recovery algorithm during matching. To search the combinatorial space of possible matches, new combinatorial optimization algorithms based upon random-start local search are utilized. These algorithms are probabilistic, and with high confidence find the globally optimal correspondence between model and image features.

The performance of four local search matching algorithms are compared on landmark recognition problems. The first algorithm uses only a form of weak perspective and, as expected, can only solve successfully a subset of the recognition problems. Because this algorithm uses a closed-form procedure to fit a 2D object model to the image it is relatively fast. The second algorithm takes a rather brute force approach to handling 3D perspective, always using an iterative 3D pose algorithm to fit the model to the data. This algorithm successfully solves all problems, but is comparatively slow. The third algorithm is a hybrid, using both the closed- form fitting of the first algorithm and the 3D pose algorithm. This algorithm successfully solves all problems and in little more time than that required by the first. The final algorithm uses a search technique called 'subset-convergence' to attempt to escape from local optima. Experiments show this algorithm to be better on what can be termed more difficult problems.

Contents

1 Introduction

The great bulk of the work on matching three-dimensional (3D) geometric object models to features extracted from two-dimensional (2D) images has not dealt quantitatively with variability in object appearance due to 3D perspective. To be more precise, algorithms have not attempted to find the best match between an object model and image features, in the sense of finding a precise 3D pose estimate such that model features projected into the image best-fit corresponding image features by some quantitative error measure. Instead, Section 3 will review how much of the work has relied upon simplifying assumptions which restrict the ways in which the appearance of an object can vary in an image. Unfortunately, such assumptions limit the effectiveness of algorithm application, and for example preclude satisfactorily solving robot landmark navigation problems such as presented later in this paper.

We have been working for several years [BWR89, BWR90] on an approach to matching based upon local search, and this approach has extended nicely to account quantitatively for full 3D perspective [BR92a, BR92b, Bev92]. A detailed account of this work may be found in Beveridge's Ph.D. dissertation [Bev93]. Matching is formalized as a combinatorial optimization task with the objective of finding the best *correspondence* between model and image features. Local search algorithms developed specifically for matching line segment features use a generate-and-test strategy to explore the combinatorial space of matches. The first step in testing any given correspondence is to fit the model to the data using a least-squares criterion.

Extending our general approach to account for 3D perspective involves replacing, or supplementing, the closed-form 2D fitting techniques presented in [BWR90] with an iterative non-linear algorithm developed by Kumar [Kum89, Kum92, KH93]. 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. Of key importance to robot navigation, assuming the camera is mounted on a mobile robot and that the object model is a landmark whose position in the world is fixed and known, this process locates the robot 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 amounts of 3D perspective. The key to this claim is the ability of our algorithms, through the integration with effective pose recovery, to take into account variations in perspective implied by different correspondence mappings. This ability can be expected to prove invaluable in a number of important application domains and it is certainly required to solve the hallway-domain landmark recognition problems presented as a test domain in this paper.

Before describing our own work, it is worthwhile to consider in broad terms the large body of work which has come before. Section 2 groups some of the significant work on model matching into four broad categories. Hopefully, readers not thoroughly familiar with the variety of previous work will find this review interesting. Following this review, Section 3 addresses 3D perspective as a topic and lays out the common assumptions made in previous work regarding permitted variability in object appearance.

2 Previous Work

Compartmentalizing a large body of work is risky. Nonetheless, a significant portion of the work on geometric object recognition falls into one of the following four broad categories:

Key-feature: Key-feature algorithms search for highly distinctive local geometric features which indicate the placement of an object in a scene; the 'local-feature-focus' method of Bolles and Cain [BC82] is an early example.

Generalized Hough: Generalized Hough algorithms [DY80, Bal81], and more recently pose clustering [Sto87] algorithms, emphasize search in the space of object model to image pose transformations.

Tree Search: Tree search algorithms [Bai85, GLP87, Gri90c, Bre90, Cas92] expand a tree of potential matches between model and image features using local geometric consistency constraints for pruning.

Geometric Hashing Geometric hashing and related approaches have arisen more recently [KSSS86, LW88]. This work can be seen as a natural extension of the key-feature approach in which local geometric features are used to predict object identity as well as placement.

Key-feature matching [BC82, Low85] covers a very broad range of specific algorithms, and is one of the more intuitive approaches. The Generalized Hough approach [DY80, Bal81] emerged at essentially the same time, although pose clustering [Sto87] was formulated later. The early contributions in tree search [Bai85, GLP87, Gri90c] came later than either key-feature or generalized Hough. Exciting extensions to tree search have recently appeared [Bre90, Cas92]. Finally, geometric hashing [KSSS86, LW88] is the most recent approach to be formulated and is the only category that explicitly addresses model indexing (selecting from among many possible objects) as well as matching.

2.1 Key-feature Matching

The general idea behind the key-feature approach is this: when looking for an object in an image, look for something both simple to find and distinctive. This is in response to the fact that, as the number of features on a model grows, the number of possible correspondence mappings between model and image features explodes exponentially. The combinatorics of search are greatly reduced by considering only a small subset of model features. As this description suggests, key-feature algorithms divide matching into two stages. In the first stage, image data is searched for instances of key-features. The second step is to verify - either accepting or rejecting - each match hypothesized by an instance of a key-feature. An early example of this approach is the 'local feature focus' method proposed by Bolles and Cain [BC82]. David Lowe [Low80, LB83, Low85] in his work on perceptual organization and recognition extends and further motivates this approach.

A difficulty with the key-feature approach is that overly simple key-features are indistinctive and hence no longer 'key', while richly distinctive key-features tend to contain more parts, and the combinatorics of searching for them grows accordingly. Sitarman and Rosenfeld [SR89] have theorized that there is some 'optimum' size for key-features. Sitarman and Rosenfeld only consider 2D-rigid models translated in the plane and hopefully future work will extend their analysis to more general problems.

Ayache and Faugeras [AF86] embody the idea of key-features in a simple control strategy in which long segments are assumed to be distinctive and hence are used as key-features. A variety of key-feature algorithms have used local curvature properties to identify object silhouettes [KJ86, GTM89, AD90]. This silhouette recognition problem is simpler than recognition in grey-scale images since the input is assumed to be a binary image.

Lowe [Low85, Low87] has made two major contributions [Low85] to the key-feature approach: 1) he argued that human perceptual organization performs feature grouping in order to find key-features which aid in recognition, and 2) he demonstrated with a working algorithm that key-features can play a significant role in recognizing 3D objects. Lowe also did a good job of quantitatively handling the problems associated with registering a 3D wire frame model to corresponding segments in an image. His early work [Low85, Low87] assumed scaled orthographic projection while his subsequent work on object tracking [Low91] handles 3D perspective.

More recent work with 3D models and an approach somewhat like the key-feature approach is that of Huttenlocher [HU90]. In the algorithm developed by Huttenlocher, feature triples are used to hypothesize matches under scaled orthographic projection. The triples are ranked according to a distinctiveness heuristic and then searched in order. This approach resembles key-feature matching in that the small, hopefully distinctive features, are searched for first. It is more prudent, in that it works down the list of potential key-features as protection against failure. This approach has been demonstrated for a variety of 3D objects.

2.2 Generalized Hough and Pose Clustering

The generalized Hough transform [DY80, Bal81, Dav82] and pose clustering [Sto87] algorithms shift the focus of search away from correspondence space and into pose space. Generalized Hough algorithms employ what are essentially voting schemes, while the more recently developed pose clustering algorithms identify tightly packed clusters of 'pose-distinctive' features. These pose distinctive features are partial matches between small numbers of model features and small numbers of image features which constrain the pose of the object model.

When there is an instance of a model in the image data, then all the pose-distinctive features associated with this instance will map to essentially the same region of pose space. Typically, generalized Hough algorithms explicitly discretize the space into cells, and then each pose-distinctive feature adds its vote to the appropriate cell. The array of cells acts as an accumulator. A cell associated with the pose of a true match should receive as many votes as there are pose-distinctive features in a model. Conceptually it is a simple matter to search the pose space for cells with sufficient votes to suggest a match. These cells, in turn, generate a hypothesized instances of the object with the indicated pose.

For a review of generalized Hough transform work up to 1988 see Illingworth and Kittler [IK88]. The principle failing of this approach is the intractability of explicitly representing pose spaces of higher than 3 dimensions. Consequently, almost all the work on generalized Hough algorithms has focused on 2D matching problems involving rigid objects. Some notable exceptions include the work of Silberberg, Harwood and Davis [SHD86], who perform a restricted form of matching in which 3D objects are assumed to be on a stable ground plane and the camera is at a known height. Thompson and Mundy [TM87] consider general unconstrained 3D views and then make judicious use of subspace projection. Specifically, votes are first projected from the six-dimensional space onto the three-dimensional subspace of rotations.

Pose clustering removes the need to explicitly represent the pose space. Stockman [Sto87] was among the first to make the case for this variation of the basic generalized Hough idea. In pose clustering, pose-distinctive features are explicitly represented as points in N-dimensional pose space, and a clustering algorithm operating over this point set identifies tightly packed dense clusters. A nice illustration of clustering local model-image feature matches based upon consistent pose hypothesis can be found in the work of Hwang [Hwa89].

Several efforts have been made to assess the strengths and weaknesses of generalized Hough matching. Brown [Bro82] wrote a comparatively early assessment of noise sensitivity. More recently, Grimson and Huttenlocher [GH88] analyzed the sensitivity of the Hough Transform and conclude that false positive rates can become intolerably high in cluttered settings or when subspace projection is used. It seems, in general, that generalized Hough and pose clustering algorithms are quite useful for low-dimensional matching problems. Their usefulness on higher-dimensional problems, such as full-perspective matching, is questionable.

2.3 Tree Search and Constraint Satisfaction

The 'tree' in tree search matching contains model-data correspondences, with the null correspondence at the root. The central idea is to expand the tree in order to find matches satisfying a set of geometric constraints. Tree search for matching geometric models to tactile sensor data was introduced by Gaston [GLP84], and was employed by Baird [Bai85] for 2D point matching. Grimson [GLP87, Gri89b, Gri89a, Gri90c] has done more than any other individual to promote tree search as a geometric matching technique, and his analytical studies [Gri90a, Gri90b, Gri90c] make tree search the best understood of the four approaches covered in this review. Recently, work by Beuler [Bre90] and Cass [Cas92] have extended Baird's ideas in some fundamental ways, leading to algorithms which intimately intertwine tree search with search in pose space.

A chronology of suggested strategies for searching the interpretation tree reflects the difficulty of problems being considered. In tactile sensing problems, the number of model and data features are typically small. Under these conditions, Gaston [GLP84] found undirected breadth-first search to be acceptable. In the case of tactile sensing, breadth-first search can be improved upon by realizing that the acquisition of data is typically sequential and expensive in terms of time. Consequently, as shown in the work of Ellis [Ell87], control strategies for tactile sensing can actively control the acquisition of sensor data based upon the degrees of freedom and possible ambiguities associated with the interpretation of the current tactile data.

Baird [Bai85] and Grimson [GLP87] favored depth-first over breadth-first search, no doubt because breadth-first search would have been extremely time consuming given the size of the problems they were studying. In addition to adopting depth-first search, Grimson cites two other factors as being important. The first is to rank choices so that the first acceptable interpretation is in some sense the 'best'. For example, data line segments can be ordered from longest to shortest, with the longest being matched first. A second important factor is to specify a stopping criterion which can be used to cut off search before a substantial portion of the interpretation tree has been expanded. An example of a stopping criterion is to call a halt to search once a consistent interpretation has been found involving data line segments whose total length exceeds a threshold.

Grimson's study of the behavior of tree search has been exemplary. These results are presented in two articles [Gri90a, Gri90b] and more fully in his book [Gri90c]. He has determined the strength of pruning based upon local geometric consistency, the importance of a stopping criterion, and the associated problems of multiple model instances and symmetric models. To summarize these results, Grimson's analysis shows 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. He also shows that without the stopping criterion, tree search is exponential in the number of consistent interpretations.

Baird's work relies upon overly restrictive assumptions about the correspondence mapping, insisting that image data contain exactly one and only one extracted image feature for every model feature. However, more recently, Beuler [Bre90] and Cass [Cas92] have extended some of Baird's ideas in ways which overcome some of the difficulties described by Grimson. Cass has coined the term *pose equivalence analysis* for his work, in which an algorithm searches simultaneously in correspondence and pose space for maximal sets of pose consistent features. The algorithm is polynomial. However, for the case involving rotation, translation and scaling of 2D models, the complexity is $O(k^4m^4d^4)$ for m model features and d data features. Here k is the number of sides bounding a polygonal uncertainty region about each feature. As Cass readily notes, although the polynomial bound is of tremendous theoretical interest, it is too high for the algorithm to be put to direct practical use.

2.4 Geometric Hashing

Not unlike the key-feature approach, geometric hashing [KSSS86, LW88, LSW90] seeks to identify features in an image that predict the presence of an object. The approach differs from key-feature matching in several important respects. Most notably, geometric hashing deals directly with multiple object models. Also, geometric hashing does not rely upon a few key-features, but instead attempts to exploit the predictive power of many subsets of image features.

Geometric hashing suffers from two basic problems: sensitivity to noise and clutter. Grimson and Huttenlocher [GH90] suggest that Lamdan's choice of metrics is quite sensitive to noise, leading to a high false positive rate even for modest amounts of noise and clutter. Costa and Shapiro [CHS90] similarly observed error sensitivity and have refined the geometric hashing approach using likelihood estimation.

To see how clutter is a problem, imagine a data set in which 1 in 5 of the data points is a clutter point. Drawing sets of 4 points at random, the probability of all four points being associated with the model is only 1 in 5. For the other 4 out of 5 point sets, looking up a model in the hash table is a waist of time and a potential source of mistakes. Now imagine if 9 out of 10 points are clutter, in which case the odds of randomly selecting 4 good points would be 1 out of $10,000$.

Others who have pushed the the underlying ideas of geometric hashing have found it necessary to put considerable effort into designing algorithms to solve the problem of selecting a set of features all belonging to same object. For instance, Stein [SM90] has shown quite nice results using a hashed indexing scheme on 2D object models. To overcome the problem of selecting features from the same object, his approach uses boundary contour information. Stein defines super segments which are consecutive sides of a polygonal boundary. A feature vector defined in terms of the relative placement of consecutive segments in the super segment is used to index object models.

More recently, Stein has extended this work to the problem of matching 3D object models to 2D image features [SM92]. Stein emphasizes the importance of grouping control mechanisms to obtain a reasonable starting set of features. Interestingly, as one who has tried to extend geometric hashing to 3D full perspective matching, he argues that this is very difficult. He instead advocates the use of topological constraints between fairly complex image features.

3 Previous Work and Full-perspective Matching

One of the most important distinctions 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*. Essentially all prior work on geometric matching has replaced *full-perspective* (the result of using a pin-hole camera model) imaging with simpler, less general, imaging models. One common imaging model is obtained by replacing perspective projection with something called scaled orthographic projection.

Using scaled orthographic projection yields what we call *weak-perspective-3D*. The '3D' is tagged onto the end to distinguish it from another weakened form of perspective imaging defined below. What is most characteristic about weak-perspective-3D imaging is that it does not induce vanishing points. In practice this means that it is an adequate approximation to full-perspective imaging for shallow objects: objects with little depth relative to their distance to the camera. However, it will increasingly distort objects for which some points on the object are significantly closer to the camera than others.

Both full-perspective and weak-perspective-3D imaging map directly from the 3D object model to the 2D image plane. Other assumptions divide the mapping process into two steps. In the first step, the 3D object model is projected, using perspective projection, into the image plane based upon a particular pose of the object relative to the camera. In the second step, this projection becomes a 2D

Table 1: Partial lists of previous work broken out by imaging model. The imaging models are defined in the accompanying text. Brief descriptions of physical circumstances leading to each are provided to aid in understanding the applicability of each.

Imaging		**Used by ...**
2D-rigid		
	Flat objects perpendicular to the camera. The distance from the camera is known.	Kalvin [KSSS86] Grimson [GLP87] Stockman [Sto87] Hwang [Hwa89] Beveridge [BWR89]
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 [AF86] Gottschalk [GTM89] Costa [CHS90] Grimson [Gri90c] Beveridge [BWR90] Cass [Cas92]
Weak-perspective-3D		
	Any 3D object shallow in depth compared to its distance from the camera and viewed from any arbitrary viewpoint. Some distortion induced for all but perfectly flat objects.	Lowe [Low85] Thompson [TM87] Huttenlocher [HU90]
Full-perspective		
	Any 3D object viewed from any arbitrary viewpoint. Full-perspective is an excellent first order approximation for a standard camera. Work listed requires an initial approximate pose estimate.	Lowe [Low91] Beveridge [BR92b]

template subjected to 2D affine transformations in the image plane. These 2D affine transformations partially account for changes in appearance associated with relative changes in object pose. In the context of matching, the first step is usually done only once prior to matching.

When a 2D similarity transform - rotation, translation and scaling - is used in place of a general affine transform, the result is a different form of weakened perspective imaging. This type of imaging shall here be called *weak-perspective-2D*. Weak-perspective-2D and weak-perspective-3D behave similarly for planar objects viewed at roughly right angles to the camera. However, these are fundamentally different forms of imaging and should not be confused. A fourth even more restrictive form of imaging is *2D-rigid*, in which only 2D rotation and translation are permitted.

Table 1 associates the four types of imaging just defined with researchers who have used each. The lists for 2D-rigid, weak-perspective-2D and weak-perspective-3D are by no means exhaustive. However, our work and that of Lowe [Low91] appear to be the only work to deal quantitatively with full 3D perspective during matching. One could argue for including also Stein's recent work [SM92], but it is omitted because Stein emphasizes qualitative over quantitative relationships when dealing with full-perspective.

There are perhaps two primary reasons why full-perspective has been avoided. First, determining the pose of an object under full-perspective is not trivial. Developing algorithms to do this in a robust and general fashion has itself been a major topic of research, and only in the past few years have Kumar [KH89, Kum92] and Lowe [Low91] developed fairly nice algorithms for pose determination.

A second reason why full-perspective has been avoided is the common reliance on local geometric constraints. These constraints are defined relative to small sets of corresponding features. The number of features required grows, and the degree of constraint shrinks, the closer the imaging model comes to full-perspective. For example, in geometric hashing [LW88], 2 points are required to establish a basis under weak-perspective-2D imaging. For full-perspective the number of points required grows to 5. This dramatically increases the computational complexity of the approach.

4 Local Search Matching Under Full 3D Perspective

Section 4.1 below briefly reviews the setup of matching as a combinatorial optimization problem and the match error used for weak-perspective-2D matching; hereafter simply called weak-perspective matching. This is followed by an overview of Kumar's 3D pose algorithm in Section 4.2. Section 4.3 introduces the robot navigation task and specific matching problems which arise in hallway navigation. The four matching algorithms tested on these problems are presented in Section 4.4 and the results of these tests are presented in Section 4.5.

4.1 The Optimal Match Minimizes 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 $d \in D$.

$$C = 2^S \qquad\qquad S \subseteq M \times D \tag{1}$$

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

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

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

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

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. 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.

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 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.

4.1.1 Fitting 2D Projections of 3D Models Assuming Weak-perspective

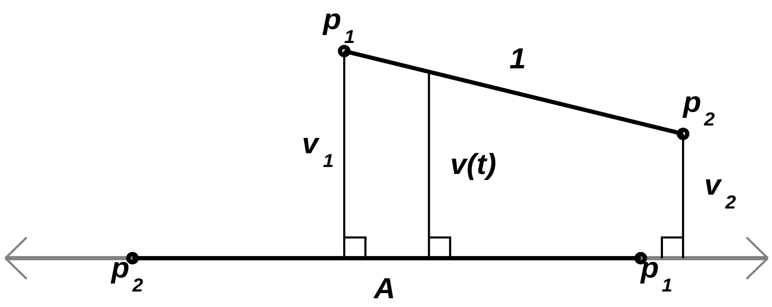


Figure 1: Points on data segment 1 project perpendicularly onto model line A. The perpendicular distance at any point along 1 may be written as a function $v(t)$.

The 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 $E_{\text{fit}}(c)$.

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:

$$v(t) \;=\; v_1 + (v_2 - v_1)\,\frac{t}{\ell} \qquad\qquad 0 \leq t \leq \ell. \tag{4}$$

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

$$\text{ISPD} \;=\; \int_0^{\ell} v^2(t)\,dt \;=\; \frac{\ell}{3}\left(v_1^2 + v_1 v_2 + v_2^2\right). \tag{5}$$

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 scaled 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.

4.2 Object Pose Under Full 3D Perspective

The basis for Kumar's algorithm [KH89] 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 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.

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].

4.3 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

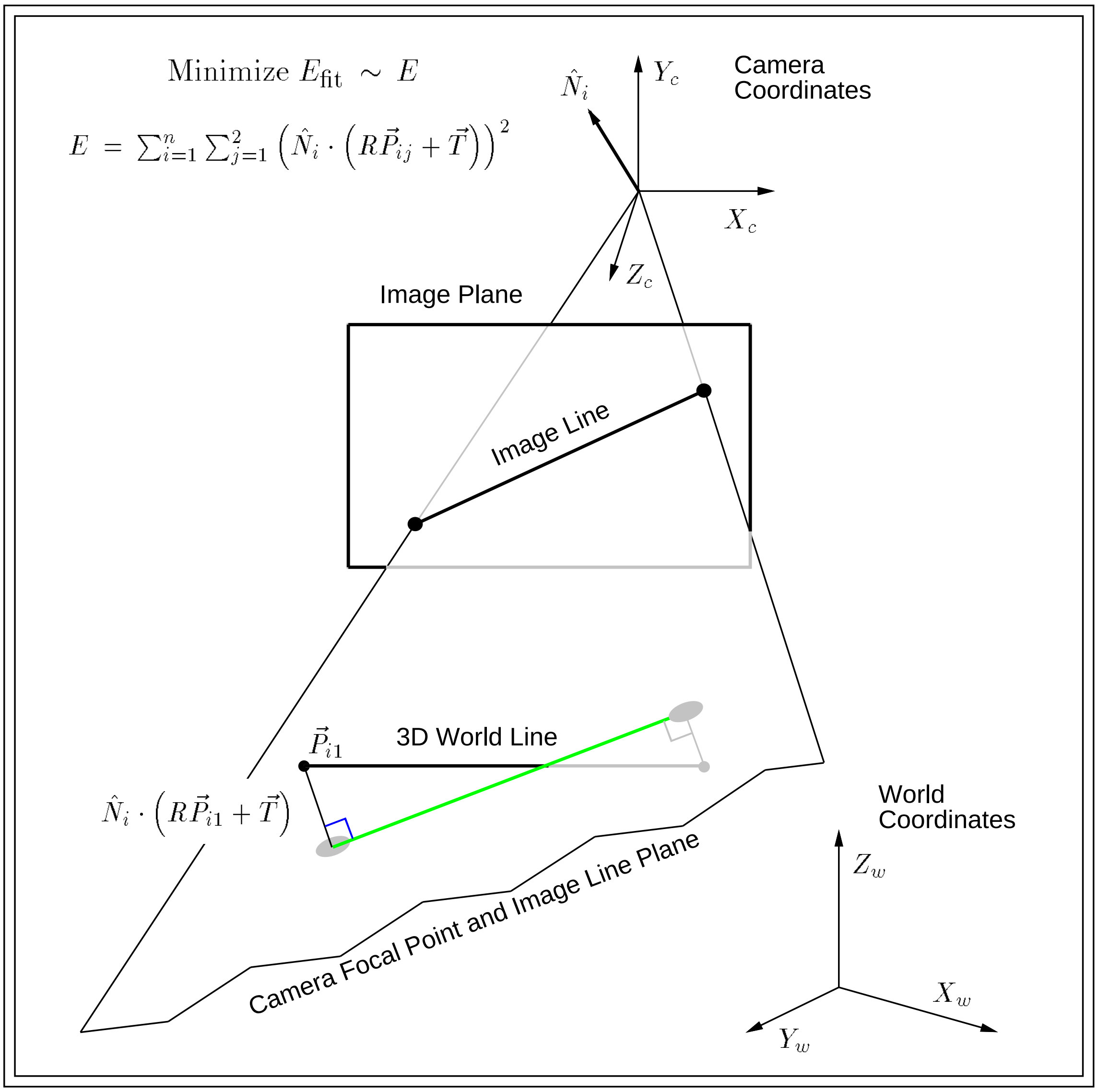


Figure 2: Point-to-plane fit measure for 3D pose algorithm.

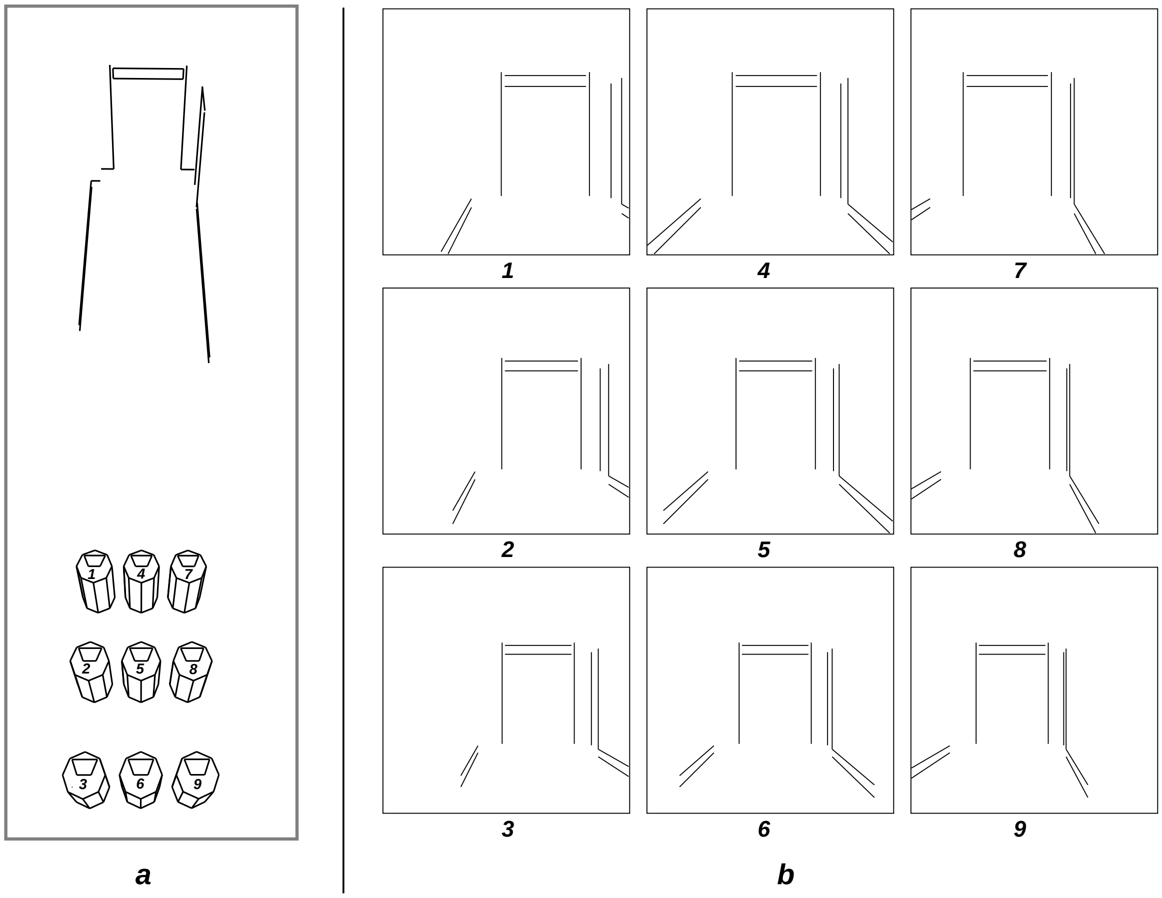


Figure 3: Perspective views of landmarks: a) Robot in relation to a partial model of the hallway. Each of the 9 poses shown are used as initial estimates from which landmark recognition tries to recover the robot's true pose. The eight poses around pose 5 are obtained by moving the robot forward and backward 4 feet and side-to-side 2 feet. 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.

(a)

(b)

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

form the basis for the experiments presented below. The goal of the experiments in Section 4.5.1 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 experiment presented in Section 4.5.2 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 problem in the second experiment is to recover the pose using the data from one image and the pose from the other as an initial estimate.

4.4 Four Matching Algorithms

Four local search matching algorithms are presented in this section. The first algorithm is a weak-perspective algorithm and the remaining three are full-perspective algorithms. The distinguishing attribute is that the full-perspective algorithms make use of Kumar's pose algorithm and therefore have the potential to solve matching problems with varying 3D perspective. In contrast, a weak-perspective algorithm can only rotate, translate and scale in 2D an initial projection of the landmark, and it cannot therefore fully account for 3D perspective.

A lengthier description of the weak-perspective algorithm, called here weak-perspective-steepest-descent, appears in [BWR90]. The three full-perspective algorithms have been reported previously in seperate papers: the full-perspective-inertial-descent algorithm in [BR92a], the hybrid-weak-full-perspective-algorithm in [BR92b], and the hybrid-subset-convergent algorithm in [Bev92]. Fuller details on all four algorithm may be found in Beveridge's Ph.D. dissertation [Bev93].

Table 2 summarizes the major attributes of the four algorithms. Some of the terms used in this table may not make sense until after the descriptions below are read. The first algorithm, weak-perspective-steepest-descent, is interesting as a baseline for performance since it does not take perspective into account. The second algorithm, full-perspective-inertial-descent, is interesting because it makes exhaustive use of Kumar's pose algorithm. The third algorithm, hybrid-weak-full-perspective, is interesting because it makes sparing use of the Kumar's pose algorithm and substantial use of the more efficient closed-form weak-perspective fitting procedure. Finally, the fourth algorithm, hybrid-subset-convergent, extends the third using a technique presented originally in [BWR90] for escaping local minima called subset convergent local search. The relative performance of these four algorithms

Table 2: Comparing the attributes of four matching algorithms. The four algorithms are presented in Sections 4.4.1, refsec:full-perspective-inertial-descent, refsec:hybrid-weak-full-perspective and 4.4.4 respectively.

		Weak-perspective-steepest-descent	Full-perspective-inertial-descent	Hybrid-weak-full-perspective	Hybrid-subset-convergent
Algorithm Type	Weak-perspective	X			
	Full-perspective		X	X	X
Pose Update Between Moves	Weak-perspective	X			
	Full-perspective		X	X	X
Local Neighborhood(s)	Hamming-distance-1	X	X	X	X
	Subset-convergent				X
Hamming-1 Neighbor Tests	Weak-perspective	X		X	X
	Full-perpective		X		
Local Search Stragetgy	Steepest-descent	X			
	Inertial-descent		X		
	Steepest-descent-tabu			X	
	Subset-convergent				X

reported below insight into the consequences of these design choices.

4.4.1 Weak-perspective Steepest-Descent Matching

As a general technique, random-starts local search combines iterative improvement and random sampling to find near optimal solutions to combinatorial optimization problems [KL72, LK73, PS82]. It is adapted to the problem of geometric matching in order to find matches which minimize E_{match}. Key to the definition of a local search algorithm is the definition of the local neighborhood. A particularly simple neighborhood may be 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. The hallway model, as projected from pose 2, is being matched to a subset of the straight line segments

Example of steepest-descent search leading to globally optimal match

		A									B												C		D		E				F		G		H	I	J
Row	**Error**	2	4	5	19	21	22	26	30	32	3	16	17	20	23	24	25	27	28	29	31	33	12	14	10	11	7	13	15	18	10	11	0	9	8	6	1
1	52.52						●							●									●	●	●		●	●	●	●	●		●	●		●	
2	13.84						●							●									●	●	●			●	●	●	●		●	●		●	
3	6.23						●							●									●	●				●	●	●	●		●	●		●	
4	4.71						●							●										●				●	●	●	●		●	●		●	
5	3.13						●							●										●				●	●		●		●	●		●	
6	0.72						●							●										●				●			●		●	●		●	
7	0.40						●							●										●				●			●			●		●	
8	0.27						●							●										●		●		●			●			●		●	
9	0.19						●							●										●		●		●			●			●	●	●	
10	0.12						●							●										●		●		●			●			●	●	●	●
11	0.09					●	●							●										●		●		●			●			●	●	●	●

Example of steepest-descent search leading to locally optimal match

		A									B												C		D		E				F		G		H	I	J
Row	**Error**	2	4	5	19	21	22	26	30	32	3	16	17	20	23	24	25	27	28	29	31	33	12	14	10	11	7	13	15	18	10	11	0	9	8	6	1
1	35.11	●	●	●	●										●	●								●	●		●				●		●		●	●	
2	13.60	●	●	●	●										●	●								●	●						●		●		●	●	
3	9.57	●	●	●	●										●	●								●	●								●		●	●	
4	3.80	●	●	●											●	●								●	●								●		●	●	
5	1.50	●	●												●	●								●	●								●		●	●	
6	1.17	●	●												●	●								●	●										●	●	
7	0.94	●													●	●								●	●										●	●	
8	0.80														●	●								●	●										●	●	
9	0.79														●	●								●	●											●	
10	0.74														●	●								●	●												
11	0.71														●									●	●												
12	0.70														●									●	●								●				

Globally optimal match | Locally optimal match

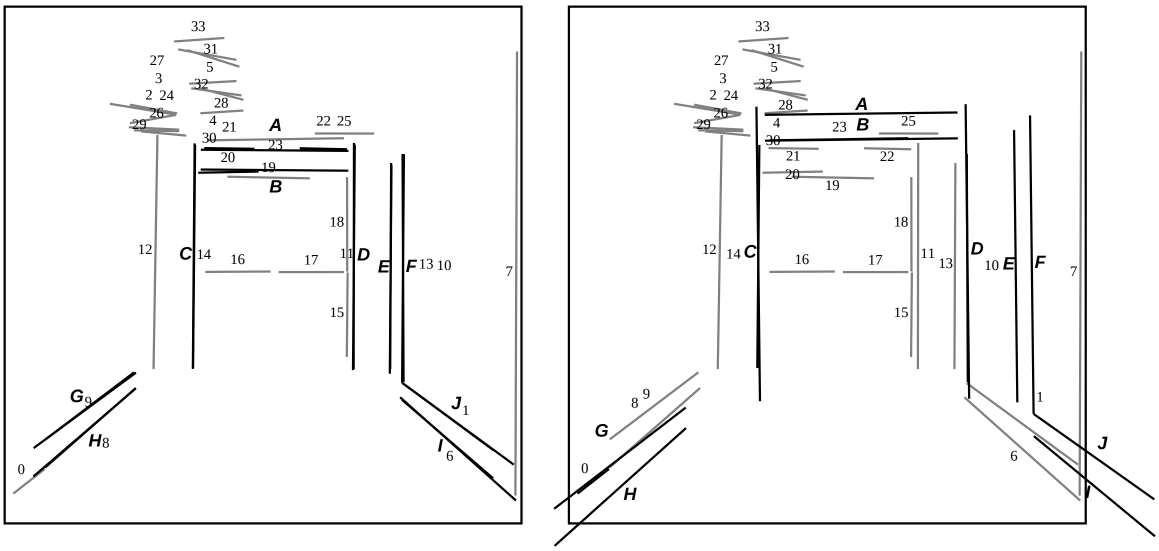


Figure 5: Steepest-descent local search using weak-perspective. The hallway model is projected from pose 2. The first row in each table indicates a correspondence chosen at random. Filled in squares indicate matched pairs. Successive rows are obtained by adding or removing the single pair yielding the greatest drop in E_{match}. In the first case the search arrives at the globally optimal match pictured on the left, and in the second case a locally optimal match is found. The grey line segments are extracted from the image shown in Figure 4a. The black line segments are projections of the 3D landmark model segments.

extracted using the Burns algorithm [BHR86] from the image shown in Figure4a. In this case weak-perspective is adequate to fit the projected model to the data.

The two tables shown in Figure 5 are interpretated as follows. 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. These candidate pairs are initially chosen based upon proximity and compatible orientation between the model and image segments. A filled in square indicates the pair is included in the current match.

By equating the entries in the rows to bits in a bit-string, the neighborhood may be equivalently described in terms of Hamming-distance. The neighborhood shown can then be described as the Hamming-distance $k = 1$ neighborhood as it consists of all correspondence mappings that differ from the current by a single bit. This Hamming-distance-1 neighborhood is used extensively by the local search algorithms presented here.

The steepest-descent strategy computes the match error for each neighboring match and then adopts the neighbor yielding the greatest improvement. In the first table, the resulting correspondence is for the globally optimal match pictured in the bottom left. In the second table, the search arrives at a locally but not globally optimal match.

One way of overcoming the problem of local optima is through random sampling, hence the use of randomly selected initial correspondences. The probability of failing to find the globally optimal match over a series of independent trials is conjunctive, hence even if the probability of success on a single trial, P_s, is fairly low, a modest number of trials is generally sufficient to be highly confident of seeing the globally optimal match. The probability of success P_s is in some sense a measure of an algorithm's strength as it has an immediate impact on its performance. Section 4.5 will begin the comparison of different algorithms by first estimating P_s for each.

4.4.2 Full-perspective-Inertial-Descent Matching

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. This means the 3D pose of the object model relative to the camera is computed

Table 3: 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	$(\,)$

for *every* match which is tested. In particular, the iterative nonlinear optimal 3D pose is run for 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 the steepest-descent strategy, n 3D poses must be computed before every step in the local search process. In an effort to save computation, a modified strategy named here *inertial-descent* is used in place of steepest-descent. The idea is to exploit information obtained when testing the n neighborhoods in order to choose a sequence of moves through the search space. 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 new, and guaranteed to be improved, match state. At this point, rather than completely evaluate the n neighbors of this new match, 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 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. Like steepest-descent, the algorithm terminates when none of the n neighbors are better than the current match.

An example of the full-perspective-inertial-descent algorithm is presented below using Figures 6 and 7 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. The model and data line segments are shown on the right hand side of Figure 6. The model line segments are shown projected into the image plane as they would appear from pose 5.

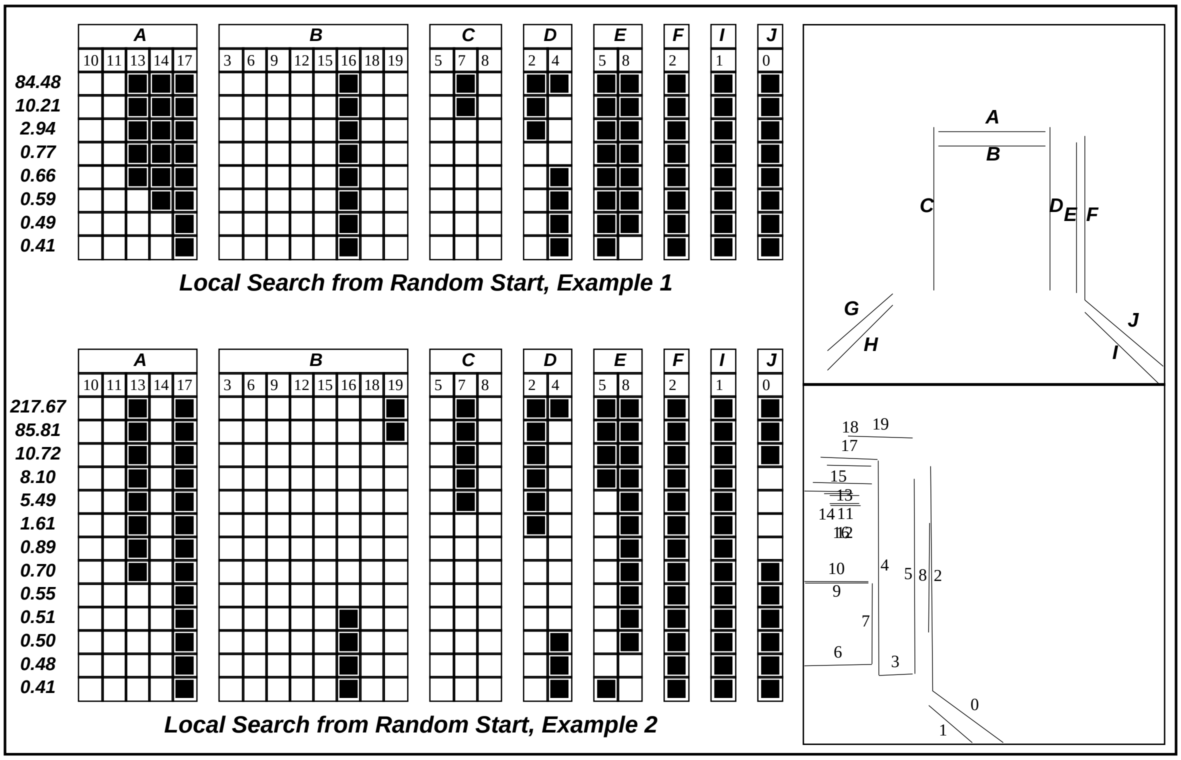


Figure 6: Example of the full-perspective-inertial-descent algorithm.

Figure 6 also shows a complete trace of two independent runs of the inertial-descent algorithm. Each successive row indicates a successively better match. Table 3 illustrates the sorted lists of transformations (bit toggles) yielding improvement for example 1 (Figure 6). 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 emphasis that new 3D poses are generated during search, Figure 7 shows the projection of the landmark from these updated poses for the successively better matches found in example 1, Figure 6. 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

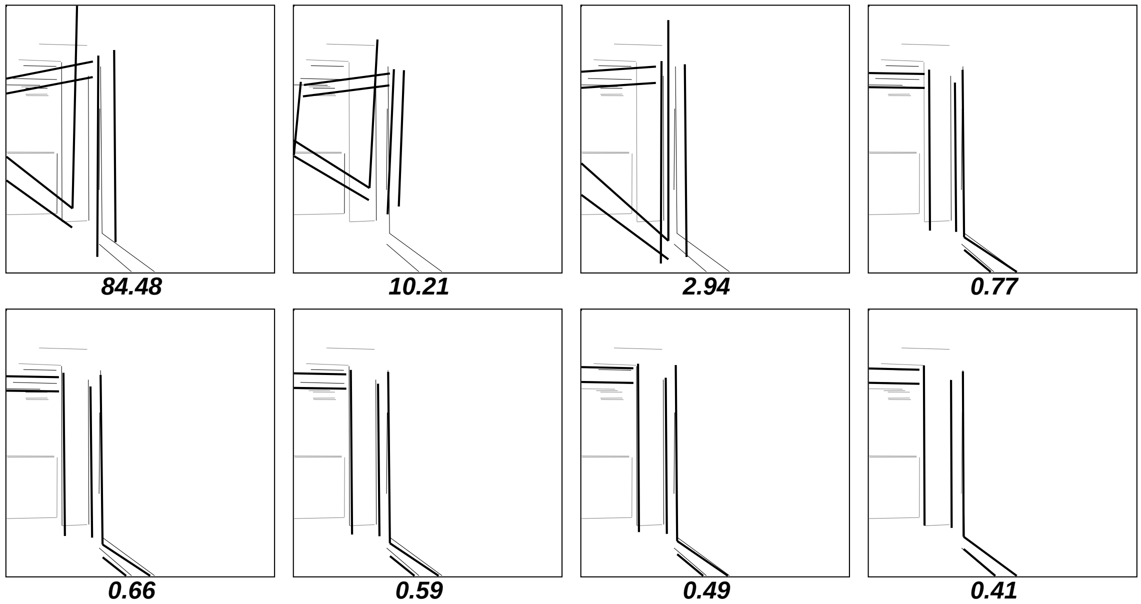


Figure 7: Landmark projections for full-perspective-inertial-descent example.

and in a room off to the right of the hallway, and of course presuming the robot could see through floors and walls. One reason to show this initial projection is to emphasize 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.

4.4.3 Hybrid Weak-perspective and Full-perspective Matching

Even using inertial-descent rather than steepest-descent, the cost of computing the 3D pose for all n neighbors makes the full-perspective-inertial-descent 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. To exploit this fact, the hybrid-weak-full-perspective algorithm developed here uses the closed-form algorithm to 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.

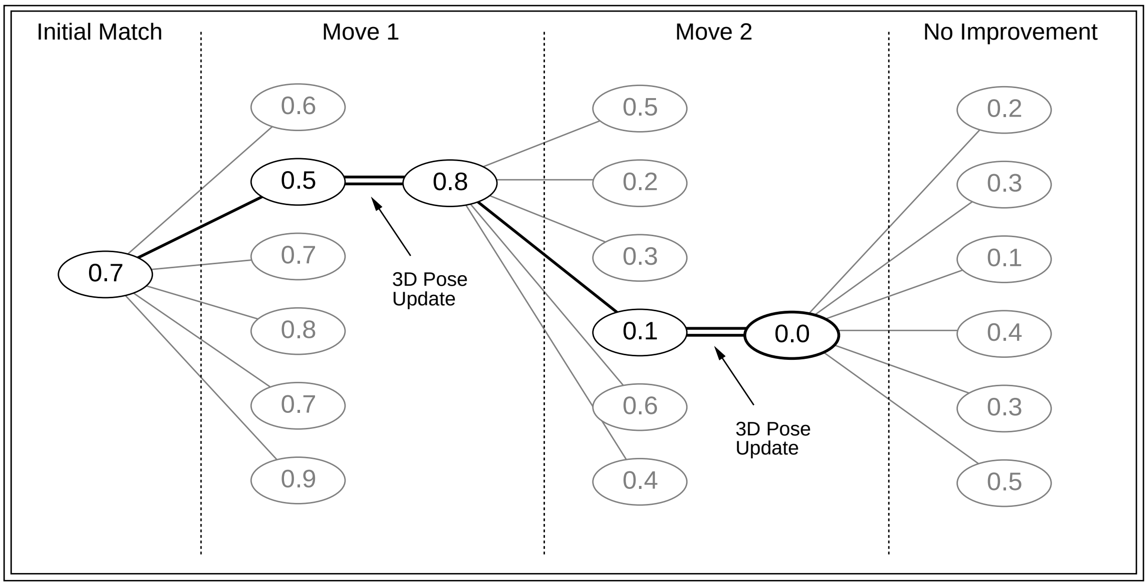


Figure 8: 3D pose update placement in hybrid-weak-full-perspective search. The nodes in the search space are shown with the weak-perspective match error E_{match} written inside. The 3D pose is updated after a best neighbor is selected. Updating pose usually alters E_{match}.

Figure 8 illustrates the essential idea of hybrid-weak-full-perspective by showing transitions to successively better match states in a search space. Nodes are drawn with the weak-perspective match error E_{match} displayed inside. These states are contrived. Their purpose is to show exactly where in the search process 3D pose is updated. To begin search, 3D pose is computed for the initial match and the model is projected into the image. Using these projected model features, the match error E_{match} is computed for all n neighbors. The neighbor with the lowest error is selected to be the next match. Search moves to this match, and in so doing, it recomputes the 3D pose and again projects the 3D model features into the 2D image, this time using the updated 3D pose estimate. This operation on the new match is indicated in Figure 8 by two bold lines connecting a node representing the selected match *prior* to the 3D pose update, and a node representing the match *after* the 3D pose update.

The possible *change* to E_{match} resulting from the 3D pose update puts a new twist on local search. Although generally E_{match} will still be lower than for the previous match, this need not always be the case. When E_{match} goes up, there are only a couple of options. One is to immediately terminate search. Another is to go back and start trying other neighbors in the hopes of finding one for which E_{match} 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, in practice it is quite common. For the hybrid-weak-full-perspective 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 must be put in to prevent search from cycling back to the already discovered best match. Cycling defeats the purpose of getting away from this match. There is a variant of local search called *tabu search* [Glo89] (Glover uses spelling 'tabu') which embodies this idea: making moves which in some sense lead backwards taboo. A transformation for the hybrid-weak-full-perspective is taboo is it adds or removes a pair of model-data features already added or removed since the last best match was found. Thus, for example, it is not possible for local search to remove a pair in one move and then undo the effect by immediately adding it back in.

4.4.4 Hybrid Subset-Convergent Matching

The hybrid-weak-full-perspective algorithm is based upon the Hamming-distance-1 neighborhood, and like the weak-perspective-steepest-descent algorithm, it can become stuck on undesirable local optima. To overcome this problem, the hybrid-subset-convergent algorithm initiates search for better matches from *subsets* of matches found to be locally optimal by the hybrid-weak-full-perspective 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. Of course the subset-convergent algorithm does not guarantee that the globally optimal match will be found on a single trial of random-start local search, but it has been shown in some cases to dramatically increase the probability of finding the optimal match on a single trial.

In keeping with all our past work, search is initiated from only *four* preselected subsets. 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 6 are the subsets used by the hybrid-subset-convergent algorithm. This coupling of subset-convergence to the hybrid-weak-full-perspective algorithm first appeared in [Bev92] and a full description of the subset-convergent control strategy for local search appears in Beveridge's Ph.D. dissertation [Bev93].

4.5 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.

4.5.1 Experiment 1: Recovering from 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 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

Table 4: 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.

	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

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 white on top and black 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. 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 4 shows the exact size of these sets for all 9 initial pose estimates.

The full-perspective-inertial-descent, hybrid-weak-full-perspective, and hybrid-subset-convergent algorithms *recovered the true pose* in all the experiments conducted for this section [1]. There were a total of 18 matching problems resulting from the 9 initial pose estimates and the two sets of candidate pairs S and $\bar{S}$. All three full-perspective 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] [2]. 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 fit the actual data. However, for the other cases the 3D pose estimates derived from the

[1] Only the position portion of the pose estimate associated with a match is considered.

[2] Speaking loosely, for these problems the 3D pose algorithm appears accurate to within roughly 6 inches.

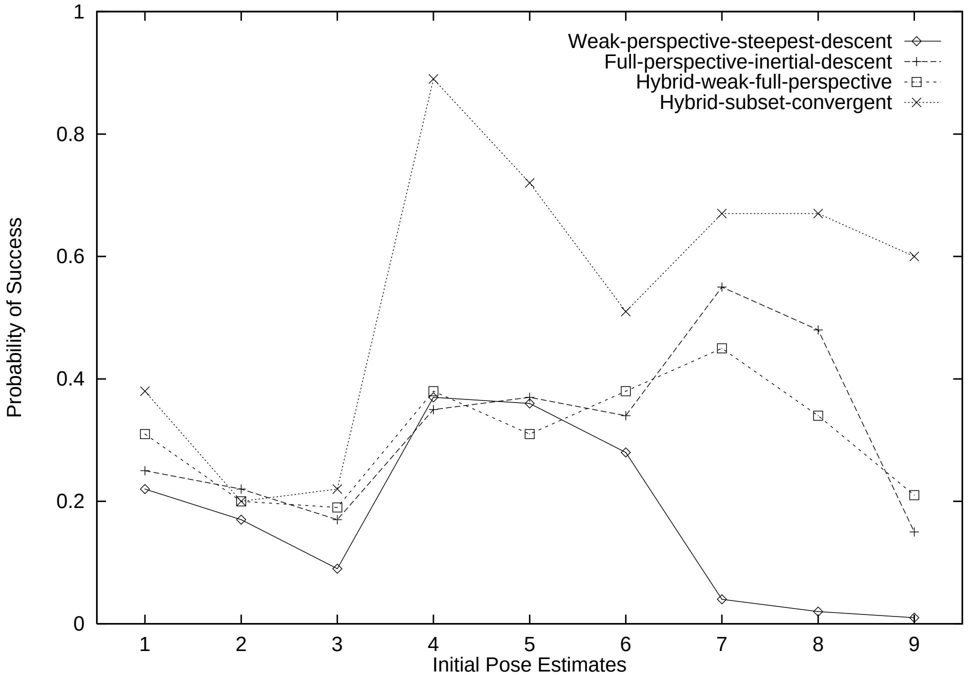
best matches differed from the true pose. They were off by 1 to 2 feet in 5 of the 6 cases, and by nearly 8 feet in one case.

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

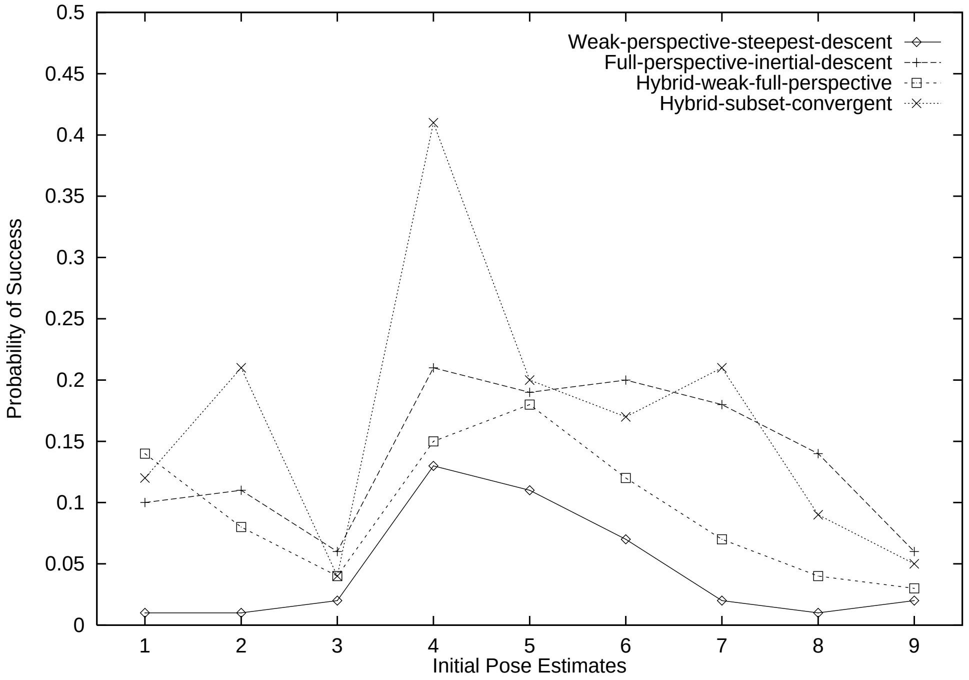
The $\hat{P}_s$ values drop when sign-of-contrast is not taken into account. As measured by $\hat{P}_s$, matching problems not using the sign-of-contrast constraint are considerably more difficult to solve. For instance, the full-perspective-inertial-descent algorithm finds the globally optimal match on a single trial with probabilities between 0.17 and 0.55 with the sign-of-contrast constraint, while without the probabilities range from 0.06 and 0.22. 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 Figure9 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. As is to be expected, the hybrid-subset-convergent algorithm outperforms the hybrid-weak-full-perspective algorithm. Finally, the weak-perspective-steepest-descent algorithm does worse than the other three in virtually all instances. Overall, this suggests that the hybrid-subset-convergent algorithm is reliably outperforming the other three in terms of the likelihood of finding the globally optimal match on a single trial.

The next set of plots, Figures 10a and 10b, show the estimated run-time, $\hat{r}_s$, required to be



(a)



(b)

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

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.

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

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. This relation is more fully developed in 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 10a is 50 minutes, while the range for the plot in Figure 10b is 5 hours. Again, it must be remembered that the search space in the latter case is incomparably larger. A new C version of this algorithm running on a Decstation 5000 is roughly 50 times faster than the Lisp Machine version reported here. When the sign-of-constrast constraint is used, this new version will never take longer than 5 seconds on these Hallway problems.

Overall, the hybrid-weak-full-perspective algorithm seems to be outperforming the others. The closest competition comes from the subset-convergent-hybrid algorithm. The worst run-time for the hybrid-weak-full-perspective algorithm is under 4 minutes using the sign-of-contrast constraint, and just under an hour otherwise. The run-times for the hybrid-subset-convergent are tending to run roughly double that required by hybrid-weak-full-perspective alone. Run-times for the full-perspective-inertial-descent algorithm are much greater, taking up to nearly an hour with the sign-of-contrast constraint, and 5 hours otherwise.

These results tell us that although the full-perspective-inertial-descent and hybrid-subset-convergent algorithms do better on individual trials, they also take more time on each trial. Moreover, neither does sufficiently better to warrant the additional time per trial. It appears, for the set of problems tested here, that the hybrid-weak-full-perspective algorithm is the best choice.

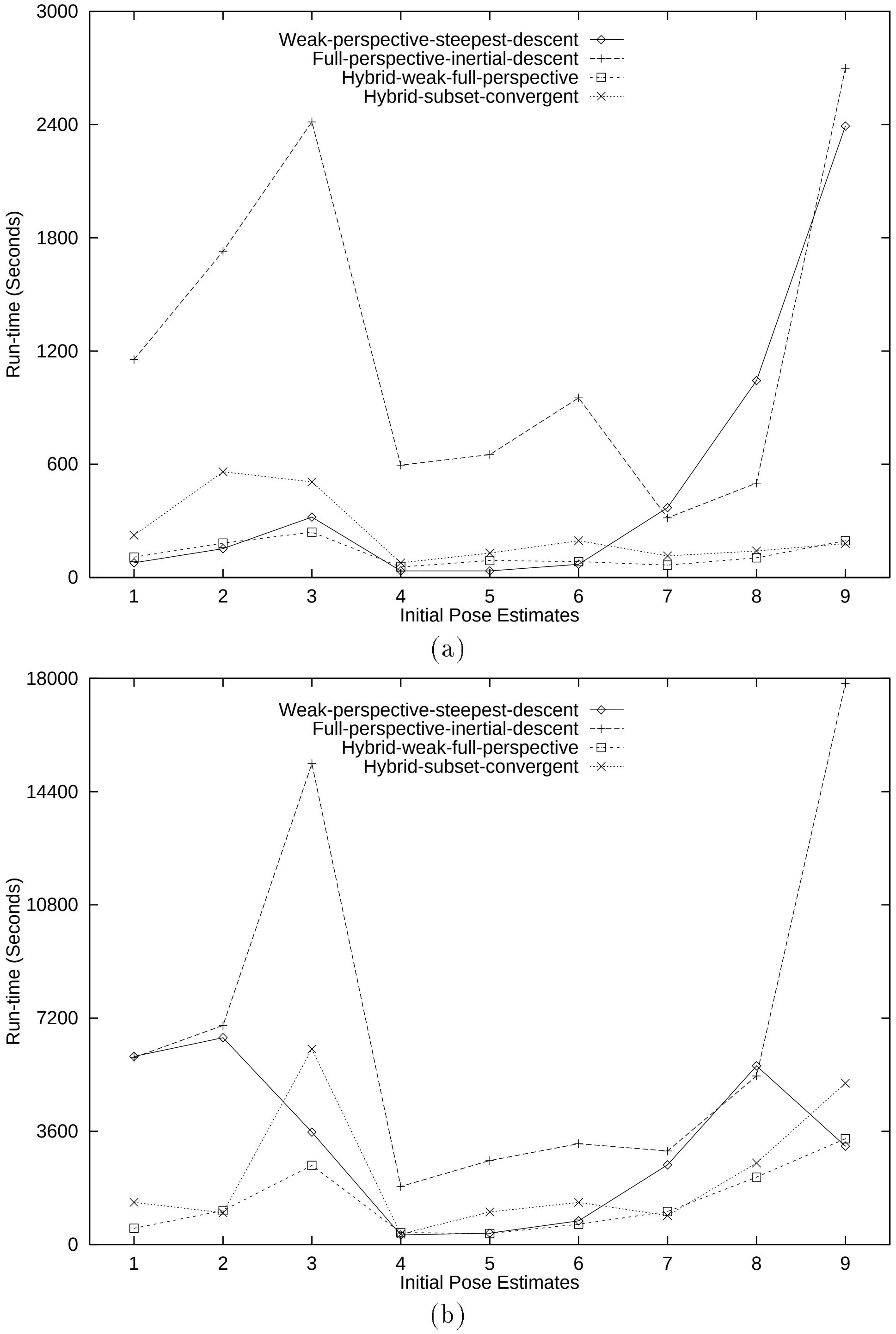


Figure 10: Estimated run-times for matching from 9 pose estimates. These times are to run sufficient trials to solve the each problem with 95% confidence. a) Using the sign-of-contrast constraint, the full-perspective-inertial-descent algorithm takes nearly an hour in the worst case. The hybrid-weak-full-perspective algorithm never takes more than 4 minutes. b) Without the sign-of-contrast constraint, the full-perspective-inertial-descent algorithm takes nearly 5 hours in the worst case. The hybrid-weak-full-perspective never takes more than an hour.

Table 5: 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.

	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-inertial-descent	0.29	172	9	1,548	0.18	63	18	1,134
Hybrid-weak-full-perspective	0.02	6	149	894	0.01	5	299	1,495
Hybrid-subset-convergent	0.09	25	32	800	0.06	11	49	539

4.5.2 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 11.

These are more difficult matching problems than those shown in the previous section, and 300 trials are used to estimate $\hat{P}_s$. Two factors make these more difficult problems. First, in one case half the expected segments are not visible while in the other many unexpected segments are visible. 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.

The results for the second experiment are summarized in Table 5. Only the three algorithms capable of handling full-perspective were tested on these problems. The table reports the estimated probability of success $\hat{P}_s$, 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. Comparing the run-times $\hat{r}_s$ using each algorithm, the hybrid-subset-convergent is superior to the other two. The highest of the two run-times for the hybrid-subset-convergent algorithm is still lower than the best time of either of the other two algorithms.

These results provide an interesting example of what is a basic tradeoff in local search: is it better to run a few expensive trials that are more likely to succeed, or instead to run many cheaper trals that are less likely to succeed? The full-perspective-inertial-descent and hybrid-weak-and-full-perspective algorithms seem to lie at opposite extremes of this tradeoff. The full-perspective-inertial-

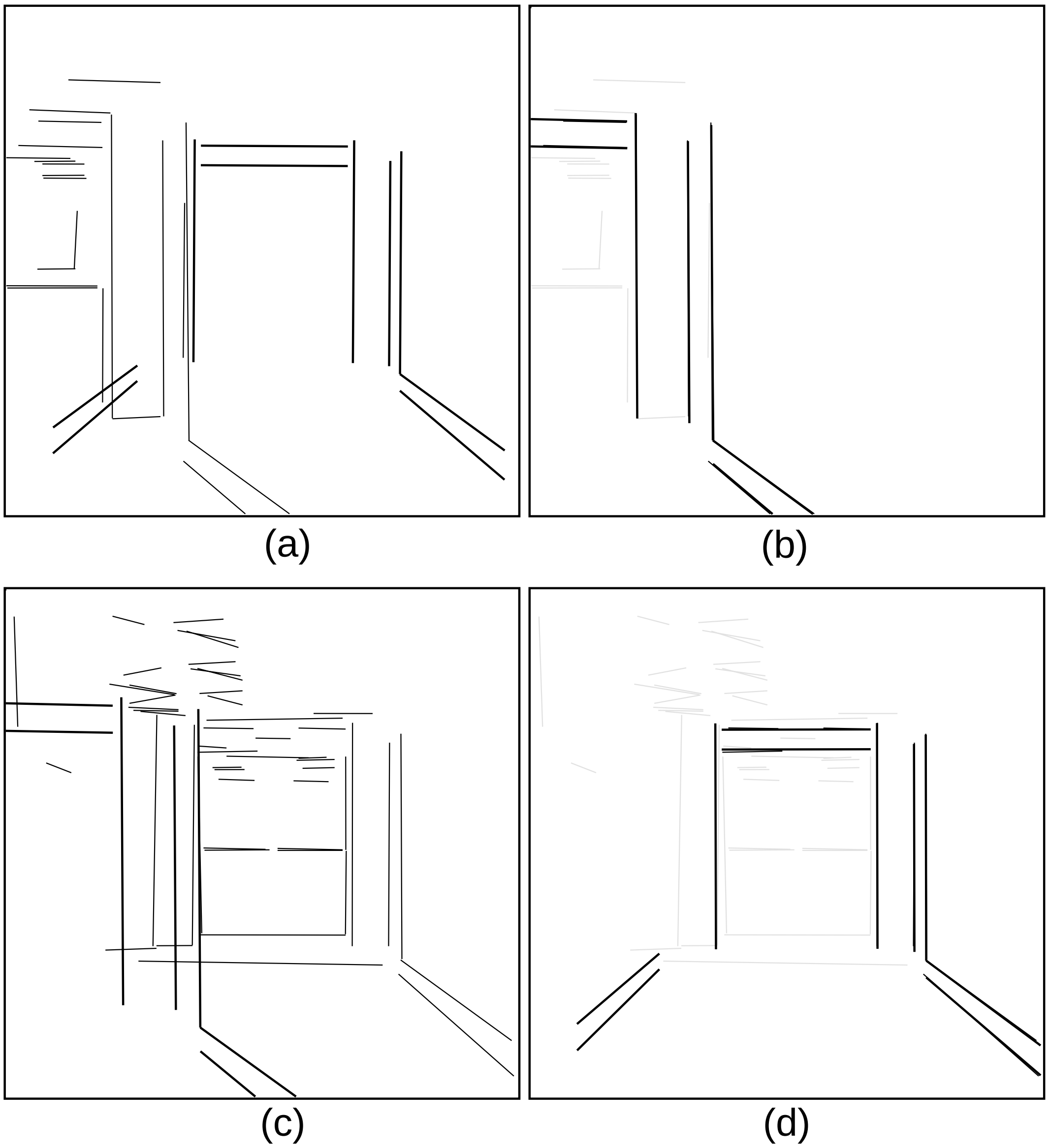


Figure 11: Confusing poses for images 1 and 2. a) landmark projected over image 2 data as it would appear from image 1 pose, b) successful recovery of correct match and pose, c) landmark projected over image 1 data as it would be seen from image 2 pose. d) successful recovery of correct match and pose.

descent algorithm is slow but reliable, finding the globally optimal match with modestly high probability on each trial, but expending considerable computation in the process. 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.

In terms of the tradeoff just mentioned, the hybrid-subset-convergent lies in between the extremes staked out by the other two. In comparing the hybrid-subset-convergent algorithm and the hybrid-weak-and-full-perspective algorithm , the increase in $\hat{P}_s$ more than makes up for the increased time taken per trial. For the hybrid-subset-convergent compared to the full-perspective-inertial-descent algorithm, each trial is sufficiently faster to make up for the relative drop in $\hat{P}_s$.

The hybrid-weak-and-full-perspective algorithm performed best on the modest pose error problems studied in the previous section, while the hybrid-subset-convergent 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 seem to suggest a trend.

5 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. By implication, the other basic approaches to geometric matching reviewed in Section 2, and which have yet to be demonstrated on perspective-sensitive matching problems, can also be expected to fail.

It was the indoor robot navigation problems presented here which provided powerful incentives to develop quantitatively accurate full-perspective matching algorithms. These algorithms are needed, in part, because often those features which most accurately determine robot pose are the same features which are most sensitive to perspective effects. This is illustrated here by the baseboards in the hallway. It is the baseboards, far more than any other features visible in the images shown in Figure 4, which must be found in order to estimate the placement of the robot left-to-right in the hallway. 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.

The experiments with different full-perspective matching algorithms have shown that it is the hybrid 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 into the initial projection of the landmark, 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 4.5.1, while the hybrid-subset-convergent version is more economical on the problems with larger initial pose error tested in Sections 4.5.2.

The experiments presented here systematically explored performance for typical hallway navigation problems, and show conclusively that the local search approach to full-perspective matching works on these problems. Determining how these results will generalize to other domains, such as outdoor navigation, will require further study. It is expected that the local search matching approach will generalize well. However, there are many details in need of further study.

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.

[Ell87] R. E. Ellis. A tactile sensing strategy for model-based object recognition. Technical Report COINS TR87-96, University of Massachusetts, Amherst, 1987.

[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.

[Gri89a] W. E. L. Grimson. On the recognition of curved objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(6):632 –643, June 1989.

[Gri89b] W. E. L. Grimson. On the recognition of parameterized 2-d objects. *Internation Journal of Computer Vision*, 2(4):353 – 372, April 1989.

[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. 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.

[Gri90c] W. Eric L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* MIT Press, Cambridge, MA, 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.

[KH93] Rakesh Kumar and Allen R. Hanson. Robust methods for estimating pose and a sensitivity analysis. *CVGIP:Image Understanding*, ?:(to appear), 1993.

[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.

[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.

[Low87] David G. Lowe. The viewpoint consistency constraint. *Iternational Journal of Computer Vision*, 1(1):58 – 72, 1987.

[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.

[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.