Robust and Efficient 3D Recognition by Alignment

by

Tao Daniel Alter

Submitted to the Department of Electrical Engineering and Computer Science
on September 8, 1992, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Alignment is a prevalent approach for recognizing three-dimensional objects in two-dimensional images. Current implementations handle errors that are inherent in images in ad hoc ways. This thesis shows that these errors can propagate and magnify through the alignment computations, such that the ad hoc approaches may not work. In addition, a technique is given for tightly bounding the propagated error, which can be used to make the recognition robust while still being efficient. Further, the error bounds can be used to formally compute the likelihood that a set of hypothesized matches between model and image features is correct.

The technique for bounding the propagated error makes use of a new solution to a fundamental problem in computer recognition, namely, the solution for 3D pose from three corresponding points under weak-perspective projection. The new solution is intended to provide a fast means of computing the error bounds. In deriving the new solution, this thesis gives a geometrical interpretation to the problem, from which the situations are inferred where the solution does not exist and is unstable.

Previous analyses of alignment have indicated that the approach is sensitive to false positives, even in moderately-cluttered scenes. But these analyses applied only to point features, whereas almost all alignment systems rely on extended features, such as line segments, for verifying the presence of a model in the image. This thesis derives a new formula for the "selectivity" of a line feature. Then, using the technique for computing error bounds, it is demonstrated experimentally that the use of line segments significantly reduces the expected false positive rate. The extent of the improvement is that an alignment system that correctly handles propagated error is expected to remain reliable even in substantially-cluttered scenes.

Thesis Supervisor: W. Eric L. Grimson
Title: Associate Professor, Department of Electrial Engineering and Computer Science

Acknowledgments

I would like to thank my advisor Eric Grimson for the direction he gave me, which was essential for completing this thesis. In addition, I have benefited immensely from conversations with my friends at the AI Lab, particularly Ronen Basri, Todd Cass, David Chanen, David Jacobs, Jose Luis Robles, Brian Subirana, Kah Kay Sung, Paul Viola, Sandy Wells, and Steve White. I am especially grateful to Jose Luis Robles, for providing great company to me innumerably many days and nights, and to my two officemates, Kah Kay Sung and Sandy Wells, for always being so helpful and so nice. I want to mention my good friends Ronen Basri and Ibrahim Hajj-ahmad, for the many memorable adventures we have had during my time at MIT. Finally, I thank my parents, Ronald and Arlene Alter, for the endless support and love they have given me my whole life, and I thank my brothers Robin and Roy, just for being the great brothers they are.

This report describes research done at the Artificial Intelligence Laboratory if the Massachusetts Institute of Technology, and was funded in part by a National Defense Science and Engineering Graduate Fellowship, and in part by the Defense Advanced Research Projects Agency of the Department of Defense under Army contract number DACA76-85-C-0010 and under Office of Naval Research contract N00014-85-K-0124.

Contents

1 Introduction **7**

1.1 Problem Definition 11

1.2 Representation 11

1.3 Approach 13

1.4 Background 15

1.5 Overview 19

2 3D Pose from 3 Points using Weak-Perspective **21**

2.1 The Perspective Case 23

2.2 Summary of 3D Pose and Direct Alignment 25

2.3 Discussion of 3D Pose 27

2.4 Existence and Uniqueness 28

2.4.1 The true solution for scale 32

2.4.2 The inverted solution for scale 33

2.5 Special Configurations of the Points 35

2.5.1 Model triangle is parallel to the image plane 35

2.5.2 Model triangle is perpendicular to the image plane 35

2.5.3 Model triangle is a line 36

2.6 Stability 37

2.7 Derivation of Direct Alignment 38
2.8 Review of Previous Solutions 41
2.9 Presentation of Three Previous Solutions 43
2.9.1 Overview 44
2.9.2 Ullman's method 45
2.9.3 Huttenlocher and Ullman's method 47
2.9.4 Grimson, Huttenlocher, and Alter's method 50
2.9.5 Summary of the three computations 53
2.10 Conclusion 55

3 Uncertainty in Point Features 57

3.1 Bounded Error Model 58
3.2 Uncertainty Circles for Bounding Uncertainty Regions 58
3.3 Cases Where Errors Are Greatest 63
3.4 Computing Uncertainty Circles 68
3.5 Expected Selectivity of Point Features 72

4 Uncertainty in Line Features 75

4.1 Line Uncertainty Regions 75
4.2 Selectivity of Line Features 76
4.2.1 Non-overlapping uncertainty circles 76
4.2.2 Overlapping uncertainty circles 82
4.2.3 Summary 85
4.3 Expected Selectivities of Line Features 86

5 Sensitivity to False Positives 89

5.1 Limits on Scene Clutter 90

5.2 Accepting a Partial Match 90
5.3 Conclusion 91

6 Likelihood of a Hypothesis 93

6.1 Formula for the Likelihood 94
6.2 Modified Formula for the Likelihood 96
6.3 Summary 98
6.4 Precomputing the Likelihoods 99
6.5 Discussion 100

7 Conclusion 101

8 Future Work 103

A Rigid Transform between 3 Corresponding 3D Points 105

B Solving for the Scale Factor 107

B.1 Biquadratic for the Scale Factor 107
B.2 Two Solutions for Scale 108
B.3 One Solution for Scale 109
B.4 No Solutions for Scale 110
B.5 Simplifying $b^2 - ac$ 110

C Generating Random Image and Model Points 113

C.1 Random Image Triples 113
C.2 Random Model Triples 114
C.3 Random Models 114

D Computing Areas of the True Uncertainty Regions 115

E Areas and Volumes of Line Uncertainty Regions **117**

E.1 True Area of a Line Uncertainty Region 117

E.2 Integrating Areas to Volumes 120

F Recurrence Relation for the Likelihood of a Hypothesis **121**

Chapter 1

Introduction

Computer vision is devoted to describing the contents of images obtained by any process that involves vision. Such processes include sensing intensity images with CCD video cameras and building depth maps using laser range-finders. Object recognition is a subfield of computer vision whose goal is to find known objects in images, such as chairs, machine parts, and people. Identifying an object as being one of a class, like "chair," turns out to be very hard. This is largely because it is difficult to describe precisely what is a chair, since chairs come in many forms and are identified partly by their shape and partly by their function. Even though it may be possible to describe a chair in terms of qualitative properties like "has a back," such descriptions are not precise enough for computer recognition.

To circumvent this problem, researchers attempt to recognize specific objects and, particularly, objects that are rigid or have rigid parts, like chairs and machine parts; Figs. 1-1 and 1-2 show some example objects. Additionally, researchers assume they are given precise *models* for the objects they wish to recognize. These models are expected to contain geometric information about the *features* on the objects, such as corners and edges. The information should include how the features are connected together and how they appear when seen from different viewpoints. Recognizing objects from such geometrically-defined models is known as "model-based" object recognition. Model-based recognition has been the paradigm for most object recognition research, and will be for this work as well.

Given a set of object models, the task is to determine which of the modeled objects are in the image, if any, and where they are. If there are not many models, recognition

Figure 1-1: Objects that are rigid or have rigid parts

Figure 1-2: Objects that are rigid or have rigid parts

can proceed by looking for the objects one at a time. Even if there is only one object to recognize as being present or not, there are several factors that make doing so difficult. The first is that an object appears differently depending on what viewpoint it is seen from, and every appearance of the object corresponding to some viewpoint must be recognized as an instance of the object. In addition, the features in images corresponding to the model contain error, due to artifacts such as inaccuracies in the imaging process, the effects of illumination, and ambiguities in feature locations. For instance, in Fig. 1-1 the poor focus and lighting make it difficult to see the 3D shape of the bookend, and the telephone edges that surround the keypad are several pixels wide. Furthermore, the object of interest may be partially occluded, or may be difficult to discern because other objects or the background look similar to it. For example, in Fig. 1-2 the telephone is partially occluded by the clamp and the flashlight, and, in addition, the back edges of the phone blend in with the white background.

One popular approach to model-based recognition that attempts to account for these problems is the "alignment" method, as described by Huttenlocher and Ullman [Huttenlocher88] [Huttenlocher90]. The general idea of alignment is to break the recognition process into two stages. The first stage uses limited information to *hypothesize* viewpoints from where an object might have been seen. For each viewpoint, the second stage computes how the model would appear in the image if seen from that viewpoint, and then examines the image to *verify* if the corresponding hypothesis is correct.

Briefly, the alignment approach uses the following mechanisms to address the problems mentioned above. To handle the fact that any view of the object could appear in the image, the method tries all possible minimal sets of matches between model and image features for hypotheses, where a minimal set contains just enough matches to compute the viewpoint from which the model was seen. To account for error in the image features, verification is performed by checking that the predicted appearance of the model matches the image only approximately. The problem of occlusion is handled by generating hypotheses using features from the model and the image that are robust to partial occlusions, such as corner points and pieces of line segments. To deal with spurious features that arise from other objects and from the background, a bottom-up process is assumed that groups together image features that are likely to come from the same object.

There are two major problems with these mechanisms, however. First, the types of features used for generating hypotheses are easily confused with similar features from other objects, from shadows, and from the background. For example, any un-modeled object in Fig. 1-1 or Fig. 1-2 can contribute spurious line segments, as can the wood-grain on the table in Fig. 1-1 and the highlights on the pencil sharpener in Fig. 1-2. This may place an excessive burden the grouping process, or alternately, may lead to a combinatorial explosion in the number of minimal sets of matches to be verified.

The second problem is with the method used to account for error in the locations of image features. The problem is that the error can propagate and magnify through the computations of the viewpoint and the appearance of the model from the viewpoint. As a result, the predicted appearance of the model may not be approximately the same as the image, but instead can be very different.

1.1 Problem Definition

In light of the mentioned problems, the objective of this thesis is to incorporate error analysis into alignment-style recognition and use the error analysis to show how to build an alignment system that is robust and efficient. As suggested above, the system is intended to recognize a restricted but wide class of 3D objects, specifically, rigid objects with sharp edges. The objects are represented by a set of geometric models, which are described in the next section. For simplicity, the system works with a small set of objects, so they can be dealt with sequentially. As input, the system is given a 2D intensity image, which may contain instances of the modeled objects. The goal is find all instances of the modeled objects in the image, or else state that none are there.

1.2 Representation

For models, the system expects to be given three data sets for each object: (1) a list of distinguished points (corners, maxima, minima, and zeros of curvature), (2) a list of extended edge features (line segments and curve segments), and (3) a complete edge description. The first data set is for generating hypotheses, the second for checking them quickly, and the third for verifying them carefully.

The third data set, the complete edge description, should consist of a small number of point-by-point, viewer-centered, 3D edge maps. The edge maps can be obtained automatically from edge-based stereo or motion, or by a laser range-finder with a 3D edge detector. In addition to shape information, the edge maps should include information about surface markings, which show up in intensity images. At this stage, there should be little thresholding on the magnitudes and lengths of edges. The edge descriptions shall be dense and noisy, but should contain enough useful information so that the object is easily identified.

About five views of each object probably are sufficient, in order to make sure at least one view covers every part of the object. More views will be necessary if an object has concavities that can only be seen from a few viewpoints; still, in this case small images should be sufficient to represent these aspects, and so not much additional storage is required. In order to predict the appearance of the object from a novel viewpoint, it may be necessary to combine information from different views. A simple way to do this is to project the entire contents of all the nearby views. The main problem with this is that the generated view may contain edges that should not be visible. On the other hand, if the edge maps contain sufficient 3D information for eliminating most hidden edges, then this problem will be minor.

The second data set, the extended edges, can be obtained by fitting relatively long straight lines and curves to the 3D edge maps. The purpose of this stage is to find model features that individually are useful for identifying the object. Compared to the density of a complete edge map, there will be very few of this type of feature.

For the extended edges, the representation is expected to be object-centered, which means that features shared by different views must be combined. Also, the viewpoints from which the features were seen should be stored with the features, so that self-occlusion can be largely accounted for. Combining features from different views may not be easy unless the views are well-registered. Even if they are not, it is possible to do this step by hand, since model-building is off-line and there are not many extended features.

As a note on smoothly curved objects, the silhouettes should not be used in obtaining the extended features, although they may give strong edges. The reason is that the silhouettes of smooth objects are not *stable*, that is, they can change as the object rotates. The extended features, on the other hand, are object-centered and may be seen from widely different viewpoints.

Finally, the features for first data set, distinguished points, can be extracted from the extended edges, either by intersecting lines or by finding extrema and inflection points on curves. There is a separate data set containing point features because they are straightforward to match between a model and an image. In contrast, extended features are likely to be broken up by the feature detector or be partially occluded. The particular point features used here (corners, extrema, and inflection points) were chosen because they are *stable*, i.e., can be identified, under projection over a wide range of views (Note double meaning "stable" here and above.)

1.3 Approach

The algorithm for recognition that I propose is as follows, and is an extension of Huttenlocher and Ullman's:

1. Form groups of image and model features, and extract distinguished points from these groups.
2. Until there are no remaining pairs of triples, hypothesize a correspondence between three grouped model points and three grouped image points.
   (a) Compute the 3D pose of the model from the three-point correspondence.
   (b) Predict the image positions of the extended features of the model using the 3D pose.
   (c) Given the error in the image points, compute a region of uncertainty for each predicted model feature that bounds the range of locations where the feature could actually lie.
   (d) Assign the three-point hypothesis a likelihood based on the uncertainty regions, using a Bayesian inference mechanism.
   (e) If the likelihood is high, select the edge maps of the model that were imaged from nearby viewpoints. Then perform a careful verification by transforming and projecting all of them into the image, which requires merging edges that are the same and eliminating edges that are hidden. Using uncertainty propagation as a guide, count how much of the projected model contour occurs in the image.
   (f) If the hypothesis verifies, remove all the distinguished image points that have been accounted for by the model from the current set of distinguished image points.
3. Return the verified hypotheses.

In this algorithm, triples of feature points are used to form hypotheses and compute poses. When such features are obtained from an image, they often come with local orientation information, which this algorithm does not make use of. For example, a feature point that is actually a corner might come with the line segments that were intersected to find the corner, or a feature point that is actually a maximum or minimum point on a curve segment, might come with an estimate of the tangent vector at that maximum or minimum. This information could in theory be used to further constrain the pose, or to do indexing (which is discussed below), so that all possible corresponding model features would not have to be examined. Although these steps would be worthwhile if properly done, it should be noted that local point features, even with orientation information, are not very distinguishing. Indexing with such features would provide a useful preprocessing step, but the brunt of the recognition problem would still remain; so this is what the above algorithm concentrates on.

For the careful-verification stage (step 2e), the edge maps that were imaged from the nearest viewpoints are used to predict the appearance of the model in the image. For most edges, this is done by transforming and projecting the edge map point-by-point. But for edges on the silhouettes of smooth objects, this does not work, since the bounding contour changes even for small rotations. For these situations, Basri and Ullman have suggested a method that can be used to bring the silhouettes into the image when the change of viewpoint is not too large [Basri88].

There are two basic differences between the algorithm listed above and Huttenlocher and Ullman's. First, Huttenlocher and Ullman's method has no formal notion of uncertainty in the feature data, whereas here handling uncertainty formally is an integral part of the algorithm. This is necessary because a small perturbation in a few point features can lead to a very different appearance of the model in the image. Although this situation could be avoided by choosing points that are far apart on the object, current grouping systems tend to locate points that are nearby. Consequently, sets of nearby points arise often and would cause a system that deals with error in an ad hoc way to break.

The second difference from Huttenlocher and Ullman's method is the use of Bayesian inference to throw away hypotheses that are unlikely. Huttenlocher and Ullman used a heuristical approach to prune hypotheses quickly. In contrast, the method here will be derived from first principles, using knowledge of how uncertainty propagates. As a result, the method here is expected to be considerably more reliable.

1.4 Background

The algorithm described in the preceding section for recognizing three-dimensional objects grew out of a number of approaches attempted in the past. Perhaps the best way to argue for its efficacy, then, is to present the development that led to its selection.

To begin, let us consider the choice of features for generating hypotheses. Early attempts used relatively large features to obtain an initial match between a model and an image. Examples of such features include convex polygons [Roberts65], projections of generalized cones [Brooks81] [Biederman85], and moments of inertia of closed regions [Cyganski85] [Reeves89]. The advantages of large features are that there are only a few of them in an image and they have few matches in the model. There is, however, a fundamental problem with these types of features: They are sensitive to partial occlusions and, as a result, cannot be extracted reliably from images.

To avoid this problem, it is common for recognition systems to extract small, local features, such as points and line segments, and then to group these together to get sets of features from the same object [Clark79] [Bolles82] [Bolles83] [Lowe85] [Thompson87] [Horaud87] [Linainmaa88] [Lamdan88a] [Huttenlocher90]. Although many systems look only for small groups of features, some of them try to find large ones. Finding large groups is a distinct problem and has received much attention [Lowe85] [Jacobs87] [Mohan88] [Horaud90] [Jacobs92]. As with large features, the advantage of large groups of features is that there are few of them in the image and the model. Ideally, a system would gather large groups of features, use them to index into a model database, and pull out exactly those models that contain features that can project to the features in that group. This approach could lead to very fast recognition, and has been examined for point features [Jacobs92].

Despite the potential gains from grouping and indexing with large groups, realistically the chances are that groups will contain spurious features and be missing correct features, and partial occlusions will make this problem much worse. In order to minimize the chance of having spurious features in groups, most systems look for groups that are small, though large enough to determine the pose of the model with respect to the data. For example, these groups can be pairs of corners [Thompson87], three-line junctions [Horaud87], triples of points [Linainmaa88] [Lamdan88a] [Huttenlocher90], and triples of lines [Clark79] [Lowe85].

Even if only small groups are available, indexing is needed to rapidly handle small to medium-sized libraries of objects. Given small groups, a model index table can be built such that all model groups which could produce a given image group can be immediately extracted. For groups containing triples of points, any model point triple can produce any image point triple under projection [Fischler81] [Huttenlocher90], and so indexing cannot help. For image groups with corners and junctions, the corresponding model groups will be somewhat constrained, but not substantially, since even these groups are not very distinguishing.

Still, it is possible to use the idea of Geometric Hashing to gain more power from indexing [Lamdan88b]. To apply Geometric Hashing to 3D recognition from 2D images, first project each model orthographically from all different points on a viewing sphere to reduce the problem to identifying flat models. Then, for each projection, take every triple of model points and, with respect to each triple, store coordinates of all the other model points in an index table, along with the model triple, the viewpoint, and the model. At recognition time, take every triple of image points and, with respect to each triple, use the coordinates of every other image point to index into the table and pull out all the model triples with those coordinates. To make the process more reliable, the look-up table should be built with points drawn from groups in the model and indexed with points drawn from large groups in the image.

Although performing indexing this way may often provide considerable filtering of hypotheses, it often will not, since, as mentioned, point features are not very distinguishing. The problem is that small sets of point features are easily confused with randomly-placed points when there is a significant amount of clutter in the scene, which means that "false positives" are likely. ([Grimson92b] gives an analysis of the likelihood of false positives in Geometric Hashing for flat objects.) The chance of false positives is further increased by taking all views of the model and, for each view, using all triples. As a consequence of false positive problems, an indexing system should be backed up with a system that tries all possible correspondences of model and image groups to find an initial match.

It may seem like a lot of work to try all possible corresponding model and image groups. For point features matched between a 3D model and a 2D image, for instance, the minimal size of a group is three [Fischler81] [Huttenlocher90]; so using points means trying all pairs of model and image triples, which, for m model points and s image points, is an $O(m^3 s^3)$ process. Nevertheless, consider again instead the possibility of using large groups. Recall that the trouble with large groups is they are not reliable enough to do indexing. Instead of using the groups for indexing, we can select triples of points from them, as was suggested for Geometric Hashing. The idea is that an unreliable group from the correct object should have at least three correct points. This would reduce the number of triples to try to a manageable level, because m would be the average number of points in a model group and s would be the average number of points in an image group. As a result, although the asymptotic complexity of $O(m^3 s^3)$ is considerable, with grouping we can expect the number of possibilities in practice to be small enough that it will be the constant time for checking a single match that decides whether the method is feasible. More generally, this is the argument that for real recognition problems, the constant factors often make the difference in whether an algorithm is efficient or not [Grimson90a].

For the reasons mentioned, bootstrapping recognition by considering all pairs of minimal sets of features is very popular. Since a minimal set of matched features is insufficient to identify an object, the minimal sets are used to find larger sets. Most techniques that do this can be divided into two broad classes, constrained search and transform clustering. Constrained search starts from each minimal hypothesis and repeatedly uses the current set of matches to constrain the search for an additional match, until a large set of matches is found [Clark79] [Brooks81] [Bolles82] [Goad83] [Grimson84] [Lowe85] [Ayache86] [Horaud87]. Transform clustering, on the other hand, uses every correspondence between a minimal set of model and image features individually to compute a model-to-image transformation, and then counts the number of times each transformation is repeated [Ballard81] [Turney85] [Thompson87] [Linainmaa88] [Cass90].

It is informative to review the motivations behind these two classes of recognition techniques. The idea of constrained search is clear, namely, to use a set of known matches to find more matches. Due to uncertainty in the positions of the features, however, this process can be difficult, since for each unmatched model feature there typically are several image features to which it can match. To handle this reliably, many systems use an extensive backtracking search [Bolles82] [Goad83] [Grimson84] [Horaud87].

The transform clustering approach avoids extensive search by noting that each correct match will independently vote for the correct transformation, so we can just let all the matches vote and then take the transformation with the most votes. Usually, this method is implemented by dividing transformation space into buckets, having each match increment a counter in a bucket, and searching the space for the buckets with the highest counts. In this form, the method is known as the "generalized Hough transform" [Ballard81].

As a recognition technique, the generalized Hough transform has several difficulties: (1) Unless the buckets are very small, the bucketing can lead to false peaks (false positives) in transformation space, since buckets combine together different transforms. (2) A more serious difficulty is that for 3D recognition from 2D images, a transformation has six degrees of freedom, which implies transformation space is six-dimensional; such a space is impractical to store and to search. For this reason, these systems do not perform a full Hough transform, but instead use separate spaces for subsets of the parameters. The effect of using separate spaces is to increase again the likelihood of false positives. (3) An additional problem is that the local features that typically are used to compute transformations between 2D images and 3D models are prone to being confused with spurious features in the image, which also can make the method prone to false positives. (4) Furthermore, it is difficult to handle uncertainty in the image features used to compute the transformation. Grimson et al. provide a way of obtaining bounds on the uncertainty in transformation space [Grimson92a], but such overestimates further increase the false positive probability. (See [Grimson90b] for an analysis of the false positive rates involved with applying the generalized Hough transform.)

For the reasons mentioned, a more reasonable use of the Hough transform is as a coarse filter to produce sets of possibly corresponding model and image features [Grimson87]. Such a stage could help considerably when looking for matches between the model and an entire image, but it may, however, not be useful if effective grouping is available.

The preceding techniques do a lot work after they are given an initial match in order to find a large set of matches. Intuitively, this seems peculiar, since, up to some uncertainty in the data, the initial match determines the pose of the model. It would seem, then, that the preceding techniques are just pinning down the model pose more precisely. As mentioned above, the reason this process is difficult is that each predicted model feature potentially matches a number of image features. Nevertheless, to resolve this ambiguity it may not be necessary to resort to constrained search or transform clustering. Instead, the ambiguity could be resolved for all model features simultaneously, by physically moving them in unison around the image. This can be done by moving the matched image features around their error regions, while continually updating the image locations of the predicted model features. The predicted model features are moved until a position is found that consistently matches most of them to within the error regions of image features. This method is equivalent to the currently-used techniques, in the sense that it will find the same set of consistent matches. At the same time, it should avoid the search through correspondence space or transformation space that they incur.

Another way to improve on earlier recognition methods is to make use of the fact that once the pose of the object is known, in theory the object's entire appearance in the image can be predicted. That is, complete edge maps could be used, instead of just a sparse set of features. This observation is the basis of Ullman's idea of using pictorial descriptions to recognize objects [Ullman89], and is one of the main ideas behind the recognition system built by Huttenlocher and Ullman [Huttenlocher88] [Huttenlocher90].

Although considerably more accurate recognition can be achieved if complete edge maps are used for verification, the expense in time of such an extensive verification would be prohibitive if it had to be done for all hypotheses. Instead, it is possible to first use the set of sparse model features to filter out a large percentage of the hypotheses. Importantly, we can do this without resolving for each predicted model feature to which of its nearby image features it corresponds. Specifically, we can compute a Bayesian estimate of the probability that a hypothesis is correct given the situation in the image (see Chapter 6). Then, once most of the hypotheses have been filtered, careful verification using complete edge maps can be performed.

In sum, a viable approach to recognition begins by locating large groups of local features in the image and the model. Then hypotheses can be formed by selecting triples of points from the groups and matching them. These matches are first checked quickly using Bayesian inference to decide how likely they are. Then they are verified carefully using detailed, viewer-centered edge maps. Also, this careful verification should be augmented to account for uncertainty in the data by trying various projections of the model.

1.5 Overview

Section 1.3 gave an algorithm for performing alignment-based recognition. The major modules of the proffered alignment algorithm are (1) grouping, (2) 3D pose computation and alignment of model features, (3) computing the likelihood of a hypothesis, and (4) careful and accurate verification.

This thesis focuses on the second and third modules. In the alignment algorithm, these modules constitute steps 2a-2d. The reason the grouping stage is not studied is that, as noted earlier, it is a distinct problem which is receiving much attention in the literature. By showing how to build a reliable recognition system independent of grouping, we may be able to infer how much is expected from a grouping stage in terms of reliable groups.

The fourth module is also a distinct problem, because it uses a different representation than the second and third modules use. In particular, the second and third modules use sparse, object-centered features, such as points and line segments, to generate hypotheses, to compute poses, and to assign likelihoods. In contrast, the fourth module uses detailed, viewer-centered edge maps to perform careful verification. In fact, the first three modules comprise an alignment system by themselves, since for many objects a sparse set of extended features is sufficient to identify them.

Chapter 2 gives a new method for computing 3D pose from three corresponding points and aligning a model to an image. The method is intended to be faster than earlier approaches, which is important because the pose computation is repeated many times. In addition, the solution is proved to be correct and is explained geometrically. Furthermore, earlier solutions to the problem are presented and compared. In addition, the stabilities of both the new and earlier solutions are analyzed.

Chapter 3 shows how to compute uncertainty regions for point features, and their selectivities. Computing the uncertainty regions quickly depends critically on the fast 3D pose computation of Chapter 2. Chapter 4 extends the analysis to line segments. Chapter 5 discusses how to use the expected selectivities for deriving formal thresholds for verification and for deciding how much scene clutter is acceptable. Chapter 6 derives a measure for ranking the hypotheses, using the selectivity formulas of Chapters 3 and 4. Lastly, Chapter 7 is the conclusion, and Chapter 8 mentions future work.

Chapter 2

3D Pose from 3 Points using Weak-Perspective

This chapter gives a new method for performing steps 2a and 2b of the alignment algorithm (Section 1.3). Specifically, this chapter shows how to compute the 3D pose of a model from three corresponding model and image points (step 2a), and how to use the pose solution to compute the image position of any unmatched model point. For step 2b, the image positions of the extended model features can be computed using points, like the endpoints of a line segment. In addition, the next chapter shows that the solution for the image position of an unmatched model point is also useful for step 2c of the alignment algorithm, in which the uncertainty regions for the predicted model features are computed. More generally, the pose solution is useful for many approaches to object recognition, such as constrained search and transform clustering (pose clustering) (Section 1.4). This is because these approaches frequently use correspondences between minimal sets of model and image features to compute poses of the model.

For computing poses of 3D objects from 2D images, a model of projection must be selected, and typically either perspective or "weak-perspective" projection is chosen. Weak-perspective projection is an orthographic projection plus a scaling, which serves to approximate perspective projection by assuming that all points on a 3D object are at roughly the same distance from the camera. The justification for using weak-perspective is that in many cases it approximates perspective closely, in particular if the size of the model in depth is small compared to the depth of the model centroid. For both perspective and weak-perspective, the minimal number of points needed to compute a model pose up to a finite number of solutions is three [Fischler81] [Huttenlocher90]. For point features, then, the problem is to determine the pose of three points in space given three corresponding image points. When perspective projection is the imaging model, the problem is known as the "perspective three-point problem" [Fischler81]. When weak-perspective is used, I shall call the problem the "weak-perspective three-point problem."

Although perspective (central) projection is a more accurate model, numerous researchers have used weak-perspective projection instead [Roberts65] [Kanade83] [Cyganski85] [CyganskiOrr88] [Thompson87] [Ullman86] [Ullman89] [Lamdan88a] [Lamdan88b] [Huttenlocher88] [Basri88] [Huttenlocher90] [Ullman91] [Jacobs91] [Grimson92a] [Grimson92b]. The reason is that there are some advantages to using weak-perspective instead of perspective. In particular, computations involving weak-perspective often are less complicated. In addition, the weak-perspective math model is conceptually simpler, since it uses orthographic instead of perspective projection. Another advantage is that we do not need to know the camera focal length or center point. Further, the effect on object recognition of errors in the image points has been studied only for weak-perspective projection [Costa90] [Jacobs91] [Lamdan91] [Rigoutsos91] [Grimson92a].

This chapter provides a new approach to recovering the pose for weak-perspective projection, which leads to a solution (method) that is intuitively simpler than earlier methods [Kanade83] [Huttenlocher87] [Cyganski88] [Huttenlocher90] [Grimson92b]. The approach here is motivated geometrically, whereas earlier methods typically are based on algebraic constraints derived from the rigidity of 3D rotations. Additionally, the geometric approach makes it easier to view what happens for special configurations of the points (Section 2.5).

A review of previous methods along with a unified presentation of their solutions is given in Sections 2.8 and 2.9. Huttenlocher and Ullman [Huttenlocher90] proved that the pose solution exists and is unique, which also is done here (Section 2.4). The solution here most resembles Ullman's [Ullman86] [Huttenlocher87], in that both end up having to solve the same biquadratic equation, although each derives the biquadratic differently. Unlike Ullman's solution, this chapter resolves which of the two non-equivalent solutions to the biquadratic is correct. Also, it explains graphically why the solutions arise and to what geometry each corresponds (Section 2.4).

In addition to providing a geomtric interpretation, the solution in this chapter leads to direct expressions for the three matched model points in camera-centered coordinates as well as an expression for the image position of any additional, unmatched model point (Section 2.7). In contrast, earlier methods all require the intermediate computation of a model-to-image transformation. Specifically, earlier solutions compute an initial transformation that brings the model into image coordinates, and then compute an additional transformation to align the matched model points to their corresponding image points. This is meaningful because many recognition systems (including the alignment algorithm in Section 1.3) calculate the 3D pose solution many times while searching for the correct pose of the model [Fischler81] [Thompson87] [Huttenlocher87] [Linnainmaa88] [Huttenlocher90] [Jacobs91] [Ullman91]. Consequently, avoiding the intermediate calculation of the transformation could cause such systems to run faster.

2.1 The Perspective Case

There is an intrinsic geometry that underlies the perspective three-point problem; it is shown in Fig. 2-1. In the figure, the three model points, $\vec{m}_0$, $\vec{m}_1$, and $\vec{m}_2$, are being perspectively projected onto three image points, $\vec{i}_0$, $\vec{i}_1$, and $\vec{i}_2$, via lines through the center of projection (center point), $\vec{p}$. The task is to recover $\vec{m}_0$, $\vec{m}_1$, and $\vec{m}_2$. The essential information is contained in the side lengths and angles of the surrounding tetrahedron.

As pictured in Fig. 2-1, I will work in camera-centered coordinates with the center point at the origin and the line of sight along the z axis. Looking at the essential parameters, the distances R_{01}, R_{02}, and R_{12} come from the original, untransformed model points. Also, the angles θ_{01}, θ_{02}, and θ_{12} can be computed from the positions of the image points, the focal length, and the center point. To see this, let f equal the focal length, and let the image points $\vec{i}_0$, $\vec{i}_1$, $\vec{i}_2$ be extended as follows: $(x, y) \rightarrow (x, y, f)$. Then

$$\cos\theta_{01} = \hat{i}_0 \cdot \hat{i}_1, \qquad \cos\theta_{02} = \hat{i}_0 \cdot \hat{i}_2, \qquad \cos\theta_{12} = \hat{i}_1 \cdot \hat{i}_2, \tag{2.1}$$

where in general $\hat{v}$ denotes the unit vector in the direction of $\vec{v}$. The problem is to determine a, b, and c given R_{01}, R_{02}, R_{12}, $\cos\theta_{01}$, $\cos\theta_{02}$, and $\cos\theta_{12}$. From the picture, we see by the law of cosines that

$$a^2 + b^2 - 2ab\cos\theta_{01} = R_{01}^2 \tag{2.2}$$
$$a^2 + c^2 - 2ac\cos\theta_{02} = R_{02}^2 \tag{2.3}$$
$$b^2 + c^2 - 2bc\cos\theta_{12} = R_{12}^2 \tag{2.4}$$

Over time, there have been many solutions to the problem, all of which start with the above equations. The solutions differ in how they manipulate the equations when solving

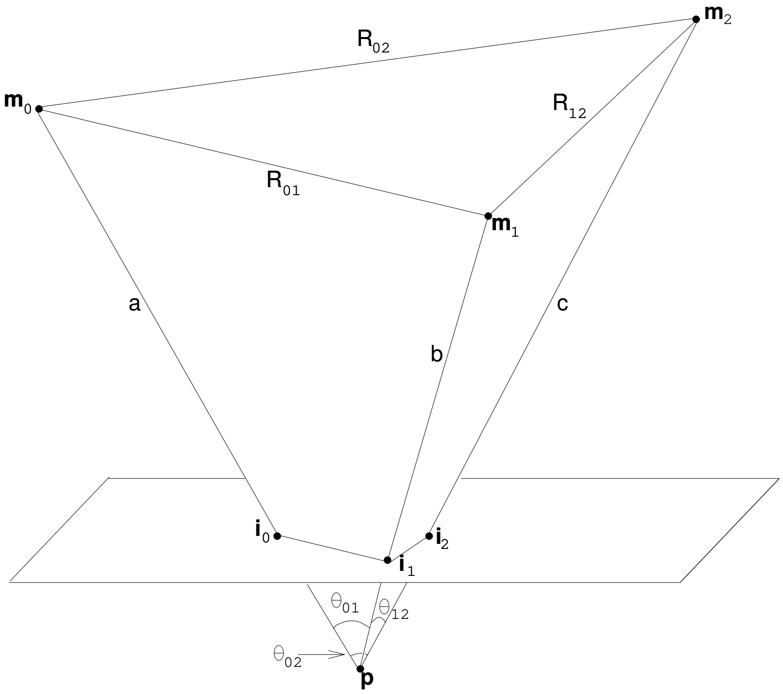

Figure 2-1: Model points $\vec{m}_0$, $\vec{m}_1$, and $\vec{m}_2$ undergoing perspective projection to produce image points $\vec{i}_0$, $\vec{i}_1$, and $\vec{i}_2$. a, b, and c are distances from the center point, $\vec{p}$, to the model points.

for the unknowns. Recently, Haralick et al. reviewed the various solutions and examined their stabilities [Haralick91].

Given a, b, and c, we easily can compute the 3D locations of the model points:

$$\vec{m}_0 = a\,\hat{i}_0 \qquad \vec{m}_1 = b\,\hat{i}_1 \qquad \vec{m}_2 = c\,\hat{i}_2. \tag{2.5}$$

If a 3D rigid transformation is desired, it can be determined from the original 3D model points and the 3D camera-centered model points just computed. A simple method for doing so is given in Appendix A; for a least-squares solution, see [Horn86].

2.2 Summary of 3D Pose and Direct Alignment

Similar to the perspective case, there is an intrinsic geometry underlying the weak-perspective three-point problem, shown in Fig. 2-2. The picture shows the three model points being projected orthographically onto the plane that contains $\vec{m}_0$ and is parallel to the image plane, and then shows them being scaled down into the image. In addition, the picture shows the model points first being scaled down and then projected onto the image plane. In each case, the projection is represented by a solid with right angles as shown. The smaller solid is a scaled-down version of the larger. The relevant information consists of the side lengths of the solids and the scale factor.

For reference, this section summarizes how to compute the locations of the three matched model points and the image location of any additional, unmatched model point. The expressions will be discussed in Section 2.3 and derived in Secs. 2.4 and 2.7. Let the distances between the model points be (R_{01}, R_{02}, R_{12}), and the corresponding distances between the image points be (d_{01}, d_{02}, d_{12}). Also let

$$\begin{aligned}
a &= (R_{01}+R_{02}+R_{12})(-R_{01}+R_{02}+R_{12})(R_{01}-R_{02}+R_{12})(R_{01}+R_{02}-R_{12}) \\
b &= d_{01}^2(-R_{01}^2+R_{02}^2+R_{12}^2)+d_{02}^2(R_{01}^2-R_{02}^2+R_{12}^2)+d_{12}^2(R_{01}^2+R_{02}^2-R_{12}^2) \\
&= R_{01}^2(-d_{01}^2+d_{02}^2+d_{12}^2)+R_{02}^2(d_{01}^2-d_{02}^2+d_{12}^2)+R_{12}^2(d_{01}^2+d_{02}^2-d_{12}^2) \\
c &= (d_{01}+d_{02}+d_{12})(-d_{01}+d_{02}+d_{12})(d_{01}-d_{02}+d_{12})(d_{01}+d_{02}-d_{12}) \\
\sigma &= \begin{cases} 1 & \text{if } d_{01}^2+d_{02}^2-d_{12}^2 \le s^2(R_{01}^2+R_{02}^2-R_{12}^2), \\ -1 & \text{otherwise.} \end{cases}
\end{aligned}$$

Then if $a \neq 0$ (otherwise see Section 2.5.3), the unknown parameters of the geometry in

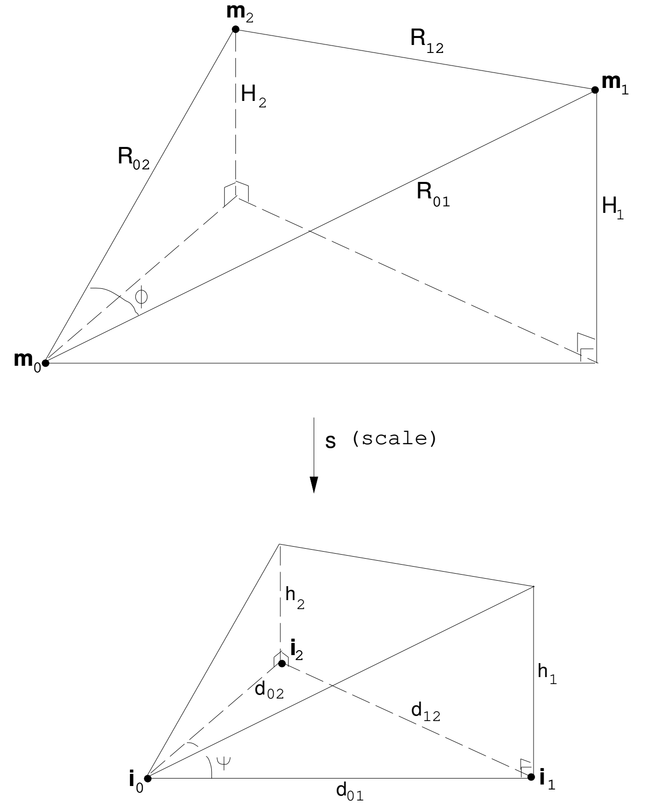

Figure 2-2: Model points $\vec{m_0}$, $\vec{m_1}$, and $\vec{m_2}$ undergoing orthographic projection plus scale to produce image points $\vec{i_0}$, $\vec{i_1}$, and $\vec{i_2}$.

Fig. 2-2 are

$$s = \sqrt{\frac{b + \sqrt{b^2 - ac}}{a}} \tag{2.6}$$

$$(h_1, h_2) = \pm\left(\sqrt{(sR_{01})^2 - d_{01}^2},\ \sigma\sqrt{(sR_{02})^2 - d_{02}^2}\right) \tag{2.7}$$

$$(H_1, H_2) = \frac{1}{s}(h_1, h_2) \tag{2.8}$$

(In practice, $|b^2 - ac|$ should be used for the inner radicand in Equation 2.6, because numerical roundoff error can cause it to become negative.)

Given image points $\vec{i}_0 = (x_0, y_0)$, $\vec{i}_1 = (x_1, y_1)$, and $\vec{i}_2 = (x_2, y_2)$, the pose solution can be used to compute the 3D locations of the model points in camera-centered coordinates:

$$\vec{m}_0 = \frac{1}{s}(x_0, y_0, w) \qquad \vec{m}_1 = \frac{1}{s}(x_1, y_1, h_1 + w) \qquad \vec{m}_2 = \frac{1}{s}(x_2, y_2, h_2 + w), \tag{2.9}$$

where w is an unknown offset in a direction normal to the image plane. It is worth noting that if the 3D rigid transform that brings the model into camera-centered coordinates is desired, it can be computed from these three camera-centered model points and the original three model points. The unknown offset w drops out when computing the rotation and remains only in the z coordinate of the translation, which cannot be recovered. As mentioned in Section 2.1, a simple method for computing the transform is given in Appendix A, and a least-squares solution is given in [Horn86].

Next, I give an expression for the image location of a fourth model point. Originally, the models points are in some arbitrary model coordinate frame. Also, the image points are in a camera-centered coordinate frame in which the image serves as the x-y plane. Denote the original, untransformed model points by $\vec{p}_i$, to distinguish them from the camera-centered model points $\vec{m}_i$ shown in Fig. 2-2. Using $\vec{p}_0$, $\vec{p}_1$, and $\vec{p}_2$, solve the following vector equation for the "extended affine coordinates," (α, β, γ), of $\vec{p}_3$:

$$\vec{p}_3 = \alpha(\vec{p}_1 - \vec{p}_0) + \beta(\vec{p}_2 - \vec{p}_0) + \gamma(\vec{p}_1 - \vec{p}_0) \times (\vec{p}_2 - \vec{p}_0) + \vec{p}_0 \tag{2.10}$$

Let $x_{01} = x_1 - x_0$, $y_{01} = y_1 - y_0$, $x_{02} = x_2 - x_0$, and $y_{02} = y_2 - y_0$. Then the image location of the transformed and projected $\vec{p}_3$ is

$$(\alpha x_{01} + \beta x_{02} + \gamma(y_{01}H_2 - y_{02}H_1) + x_0,\ \alpha y_{01} + \beta y_{02} + \gamma(-x_{01}H_2 + x_{02}H_1) + y_0). \tag{2.11}$$

2.3 Discussion of 3D Pose

Section 2.4 will show the following results, in addition to deriving the 3D pose solution given in the last section. The pose solution has a two-way ambiguity unless h_1 and h_2 are zero (Equation 2.7). The ambiguity corresponds to a reflection about a plane parallel to the image plane. When $h_1 = h_2 = 0$, the model triangle (the triangle defined by the three model points) is parallel to the image triangle (the triangle defined by the three image points). As a note, a and c measure sixteen times the squares of the areas of the model and image triangles, respectively. Further, the solution fails when the model

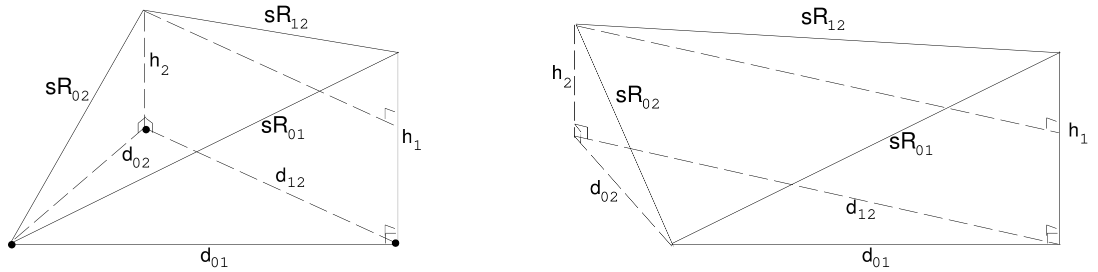

Figure 2-3: Orthographic projection plus scale of a model triangle into an image.

triangle degenerates to a line, in which case $a = 0$; in fact, this is the only instance in which a solution may not exist (Section 2.5.3). Note that no such restriction is placed on the image triangle; so the image points may be collinear. Note also that no restriction is placed on the shape of the triangles, although the triangles in Fig. 2-2 are acute. For illustration, Fig. 2-3 right shows a picture for when the model triangle is acute and the image triangle is not, along with the smaller solid from Fig. 2-2.

Next, notice that all that is pertinent to recovering the 3D pose of the model are the distances between the model and image points, not their locations. Previous solutions have used the actual locations of the points to compute the pose, after first applying a rigid transformation to put the three model points in the image plane [Huttenlocher87] [Huttenlocher90] [Grimson92a].

In terms of the ordering of the points, the symmetry in Equations 2.6-2.6 shows that the scale factor is the same, independent of the ordering. Previous methods that are based on the coordinates of the points after some initial transformations make this symmetry unclear. For the altitudes H_1 and H_2 (or h_1 and h_2), we can see from Fig. 2-2 how the different orderings are related: In Fig. 2-2 the solution is based at $\vec{m}_0$, and the altitudes are H_1 for $\vec{m}_1$, H_2 for $\vec{m}_2$, and 0 for $\vec{m}_0$. For a solution based at $\vec{m}_1$, the altitudes become 0 for $\vec{m}_1$, $H_2 - H_1$ for $\vec{m}_2$, and $-H_1$ for $\vec{m}_0$. For a solution based at $\vec{m}_2$, the altitudes become $H_1 - H_2$ for $\vec{m}_1$, 0 for $\vec{m}_2$, and $-H_2$ for $\vec{m}_0$.

2.4 Existence and Uniqueness

This section derives the 3D pose solution and shows that the solution exists for all sets of model and image points except when the model points are collinear, and that the solution is always unique. In deriving the 3D pose solution, I start with the basic geometry for the weak-perspective three-point problem, shown in Fig. 2-3. There are three right triangles in each solid, from which three constraints can be generated:

$$h_1^2 + d_{01}^2 = (sR_{01})^2 \tag{2.12}$$
$$h_2^2 + d_{02}^2 = (sR_{02})^2 \tag{2.13}$$
$$(h_1 - h_2)^2 + d_{12}^2 = (sR_{12})^2 \tag{2.14}$$

The distances R_{01}, R_{02}, R_{12}, d_{01}, d_{02}, d_{12} and the scale factor s are all positive, but the altitudes h_1, h_2 along with H_1, H_2 are signed. Since h_1 and h_2 are signed, having "$h_1 - h_2$" in the third equation is an arbitrary choice over "$h_1 + h_2$"; it was chosen because, when h_1 and h_2 are positive, it directly corresponds to the pictures in Fig. 2-3.

Multiplying the third equation by -1 and adding all three gives

$$2h_1h_2 = s^2(R_{01}^2 + R_{02}^2 - R_{12}^2) - (d_{01}^2 + d_{02}^2 - d_{12}^2). \tag{2.15}$$

Squaring and using the first two equations again to eliminate h_1^2 and h_2^2, we have

$$4(s^2R_{01}^2 - d_{01}^2)(s^2R_{02}^2 - d_{02}^2) = \left(s^2(R_{01}^2 + R_{02}^2 - R_{12}^2) - (d_{01}^2 + d_{02}^2 - d_{12}^2)\right)^2, \tag{2.16}$$

which, after some manipulation, leads to a biquadratic in s (for details see Appendix B.1):

$$as^4 - 2bs^2 + c = 0, \tag{2.17}$$

where

$$\begin{aligned}
a &= 4R_{01}^2R_{02}^2 - (R_{01}^2 + R_{02}^2 - R_{12}^2)^2 \\
&= (R_{01} + R_{02} + R_{12})(-R_{01} + R_{02} + R_{12})(R_{01} - R_{02} + R_{12})(R_{01} + R_{02} - R_{12}) \\
b &= 2R_{01}^2d_{02}^2 + 2R_{02}^2d_{01}^2 - (R_{01}^2 + R_{02}^2 - R_{12}^2)(d_{01}^2 + d_{02}^2 - d_{12}^2) \\
&= d_{01}^2(-R_{01}^2 + R_{02}^2 + R_{12}^2) + d_{02}^2(R_{01}^2 - R_{02}^2 + R_{12}^2) + d_{12}^2(R_{01}^2 + R_{02}^2 - R_{12}^2) \\
&= R_{01}^2(-d_{01}^2 + d_{02}^2 + d_{12}^2) + R_{02}^2(d_{01}^2 - d_{02}^2 + d_{12}^2) + R_{12}^2(d_{01}^2 + d_{02}^2 - d_{12}^2) \\
c &= 4d_{01}^2d_{02}^2 - (d_{01}^2 + d_{02}^2 - d_{12}^2)^2 \\
&= (d_{01} + d_{02} + d_{12})(-d_{01} + d_{02} + d_{12})(d_{01} - d_{02} + d_{12})(d_{01} + d_{02} - d_{12})
\end{aligned}$$

In Fig. 2-2, let ϕ denote the angle between $\vec{m}_1 - \vec{m}_0$ and $\vec{m}_2 - \vec{m}_0$, and let ψ be the angle between $\vec{i}_1 - \vec{i}_0$ and $\vec{i}_2 - \vec{i}_0$. Notice by the law of cosines that

$$\begin{aligned}
a &= 4R_{01}^2R_{02}^2 - (2R_{01}R_{02}\cos\phi)^2 \quad = \quad 4(R_{01}R_{02}\sin\phi)^2 && (2.18)\\
b &= 2R_{01}^2d_{02}^2 + 2R_{02}^2d_{01}^2 - (2R_{01}R_{02}\cos\phi)(2d_{01}d_{02}\cos\psi) \\
&= 2(R_{01}^2d_{02}^2 + R_{02}^2d_{01}^2 - 2R_{01}R_{02}d_{01}d_{02}\cos\phi\cos\psi) && (2.19)\\
c &= 4d_{01}^2d_{02}^2 - (2d_{01}d_{02}\cos\psi)^2 \quad = \quad 4(d_{01}d_{02}\sin\psi)^2 && (2.20)
\end{aligned}$$

Further, $\frac{1}{2}R_{01}R_{02}\sin\phi$ equals the area of the model triangle, so that a measures sixteen times the square of the area of the model triangle. Analogously, c measures sixteen times the square of the area of the image triangle.

The biquadratic in Equation 2.17 is equivalent to the one originally derived by Ullman. But Ullman made no attempt to interpret or decide among its solutions, which will be done here. We are interested only in positive, real solutions for s, the scale factor. In general, the positive solutions of the biquadratic are given by

$$s = \sqrt{\frac{b \pm \sqrt{b^2 - ac}}{a}} \qquad (2.21)$$

Depending on the radicands, there will be zero, one, or two real solutions. Particularly, we are interested in whether each number of solutions can arise, and, if so, to what the solutions correspond geometrically.

In what follows, I assume that the model triangle is not degenerate, that is, not simply a line or a point. This situation is the only time the solution is not guaranteed to exist (see Section 2.5.3). Note that this assumption implies that $a \neq 0$ and $\phi \neq 0, \pi$.

To begin, let us determine the signs of a, b, and c. From Equations 2.18 and 2.20, clearly $a > 0$ and $c \geq 0$. From Equation 2.19, it is straightforward to see that $b > 0$, since

$$\begin{aligned}
b &= 2(R_{01}^2d_{02}^2 + R_{02}^2d_{01}^2 - 2R_{01}R_{02}d_{01}d_{02}\cos\phi\cos\psi) \\
&> 2(R_{01}^2d_{02}^2 + R_{02}^2d_{01}^2 - 2R_{01}R_{02}d_{01}d_{02}), \quad \text{since} \cos\phi < 1,\ \cos\psi \leq 1 \\
&= 2(R_{01}d_{02} - R_{02}d_{01})^2 \quad \geq \quad 0
\end{aligned}$$

Via some algebra (given in Appendix B.5), it can be shown that

$$b^2 - ac = 4(R_{01}d_{02})^4\left(t^2 - 2\cos(\phi+\psi)t + 1\right)\left(t^2 - 2\cos(\phi-\psi)t + 1\right), \qquad (2.22)$$

where $t = \frac{R_{02}d_{01}}{R_{01}d_{02}}$, which leads to $b^2 - ac \geq 0$ (see Appendix B.4). From this fact and that $a > 0$, $b > 0$, and $c \geq 0$, we can derive that there are in general two solutions for s with a single special case when $b^2 - ac = 0$, which can be seen as follows:

$$\begin{aligned} b^2 - ac \geq 0 \quad &\Longrightarrow \quad b \pm \sqrt{b^2 - ac} \geq 0, \text{ since } b > 0 \text{ and } ac \geq 0 \\ &\Longrightarrow \quad \frac{b \pm \sqrt{b^2 - ac}}{a} \geq 0, \text{ since } a > 0 \end{aligned}$$

Hence $s = \sqrt{\frac{b \pm \sqrt{b^2 - ac}}{a}}$, which gives one or two solutions for the biquadratic, depending on whether $b^2 - ac$ is equal to zero or is positive.

Next I show that of the two solutions for the scale, exactly one of them is valid, that is, corresponds to an orthographic projection of the model points onto the image points. Furthermore, the other solution arises from inverting the model and image distances in Fig 2-2. In addition, there being one solution for scale corresponds to the special case in which the model triangle is parallel to the image plane. The following proposition will be used to establish these claims.

Proposition 1: Let

$$s_1 = \sqrt{\frac{b - \sqrt{b^2 - ac}}{a}} \qquad s_2 = \sqrt{\frac{b + \sqrt{b^2 - ac}}{a}}. \tag{2.23}$$

Then

$$s_1 \leq \frac{d_{01}}{R_{01}}, \frac{d_{02}}{R_{02}} \leq s_2. \tag{2.24}$$

Proof: s_1 and s_2 are solutions to the biquadratic in Equation 2.17. Since $a > 0$, the quadratic function in s^2 on the left-hand side of Equation 2.17 is concave up and, consequently, is negative exactly in the interval between the zeroes s_1^2 and s_2^2. Further, by substitution it can be seen that this function takes on negative values for $s^2 = \left(\frac{d_{01}}{R_{01}}\right)^2$ and $s^2 = \left(\frac{d_{01}}{R_{01}}\right)^2$ (see Appendix B.2). Since the scale factors and the distances are non-negative, this immediately gives that $\frac{d_{01}}{R_{01}}$ and $\frac{d_{02}}{R_{02}}$ lie between s_1 and s_2.
$\square$

2.4.1 The true solution for scale

Here it is shown that exactly one of the two solutions for scale can satisfy the geometry shown in Fig. 2-2, and it is always the same one. If the two solutions are the same, then both solutions can satisfy the geometry (this case is discussed in Section 2.5.1). As will be seen, the valid solution is

$$s = s_2 = \sqrt{\frac{b + \sqrt{b^2 - ac}}{a}}\,. \tag{2.25}$$

Note that proving this statement establishes the existence and uniqueness of the pose solution.

In Fig. 2-3, $(sR_{01})^2 - d_{01}^2 = h_1^2 \geq 0$ and $(sR_{02})^2 - d_{02}^2 = h_2^2 \geq 0$, which implies that any solution s satisfies $\frac{d_{01}}{R_{01}} \leq s$ and $\frac{d_{02}}{R_{02}} \leq s$. Consequently, Proposition 1 implies that s_2 is the only possible solution.

The question remains whether s_2 is itself a solution. The fact that it satisfies the biquadratic is not sufficient since the squaring used to obtain Equation 2.16 from Equation 2.15 may not be reversible. Yet we do know s_2 satisfies Equation 2.16, because the steps from Equation 2.16 to Equation 2.21 are reversible. Consequently, Equation 2.15 will be satisfied if the sign of h_2 relative to h_1 is chosen accordingly. Let σ be the sign of h_2 when the sign of h_1 is 1, and $-\sigma$ be h_2's sign when h_1's sign is -1. Then unless the right-hand side of this equation is 0, Equation 2.15 is satisfied by

$$\sigma = \begin{cases} 1 & \text{if } d_{01}^2 + d_{02}^2 - d_{12}^2 < s^2(R_{01}^2 + R_{02}^2 - R_{12}^2), \\ -1 & \text{if } d_{01}^2 + d_{02}^2 - d_{12}^2 > s^2(R_{01}^2 + R_{02}^2 - R_{12}^2). \end{cases}$$

If on the other hand $s^2(R_{12}^2 - R_{01}^2 - R_{02}^2) = d_{12}^2 - d_{01}^2 - d_{02}^2$, then Equation 2.15 implies h_1 or h_2 is 0, so that the sign of h_2 relative to h_1 is arbitrary.

Notice that the collective sign of h_1 and h_2 is still free, and so there is a two-way ambiguity in the pairs (h_1, h_2) and (H_1, H_2). As can be seen in Fig. 2-2, the ambiguity geometrically corresponds to a flip of the plane containing the space points $\vec{m}_0$, $\vec{m}_1$, and $\vec{m}_2$. The flip is about a plane in space that is parallel to the image plane, but which plane it is cannot be determined since the problem gives no information about offsets of the model in the z direction. Due to the reflection, for planar objects the two solutions are equivalent, in that they give the same image points when projected. On the other hand, for non-planar objects the two solutions project to two different sets of points.

There is a special case, as mentioned above, when the sign of h_2 is arbitrary relative to the sign of h_1. In this case, the right-hand side of Equation 2.15 is zero, and this implies that h_1 or h_2 is zero also. Looking at Fig. 2-2, geometrically what is occurring is that one of the sides of the model triangle that emanates from $\vec{m}_0$ lies parallel to the image plane, so that the reflective ambiguity is obtained by freely changing the sign of the non-zero altitude.

2.4.2 The inverted solution for scale

Of the two solutions for scale that satisfy the biquadratic, we know that s_2 corresponds to the geometry in Fig. 2-2, but what about s_1? Using a similar argument to that used to prove s_2 is a solution for the weak-perspective geometry, we can infer a geometric interpretation for s_1. Consider, then, $s = s_1$. The interpretation I will derive satisfies the equations,

$$\begin{aligned} H_1^2 + R_{01}^2 &= (rd_{01})^2 && (2.26)\\ H_2^2 + R_{02}^2 &= (rd_{02})^2 && (2.27)\\ (H_1 - H_2)^2 + R_{12}^2 &= (rd_{12})^2, && (2.28) \end{aligned}$$

where $r = \frac{1}{s}$. Observe that $r = \frac{1}{s_1}$ and s_2 have similar forms (compare to Equation 2.25):

$$r = \sqrt{\frac{a}{b - \sqrt{b^2 - ac}}} = \sqrt{\frac{b + \sqrt{b^2 - ac}}{c}}. \qquad (2.29)$$

To begin the derivation, Proposition 1 gives that $d_{01}^2 - (sR_{01})^2 \geq 0$ and $d_{02}^2 - (sR_{02})^2 \geq 0$, which implies we can set $h_1^2 = d_{01}^2 - (sR_{01})^2$ and $h_2^2 = d_{02}^2 - (sR_{02})^2$. Dividing through by s^2 gives Equations 2.26 and 2.27. Since s_1 satisfies Equation 2.16 (for the same reason s_2 did), we can substitute into Equation 2.16 with h_1^2 and h_2^2 to obtain $(h_1 - h_2)^2 = d_{12}^2 - s^2 R_{12}^2$, where the sign of h_2 relative to h_1 is 1 if $d_{01}^2 + d_{02}^2 - d_{12}^2 \geq s^2(R_{01}^2 + R_{02}^2 - R_{12}^2)$, and -1 otherwise. Dividing through by s^2 gives Equation 2.28, and so the derivation is completed.

Geometrically, Equation 2.26 forms a right triangle with sides H_1 and R_{01}, and hypotenuse rd_{01}. Analogously, Equations 2.27 and 2.28 imply right triangles as well. The interpretation is displayed in Fig. 2-4. Another way to see what is occurring geometrically is to note that the roles of the model and image distances from Equations 2.12-2.14 are

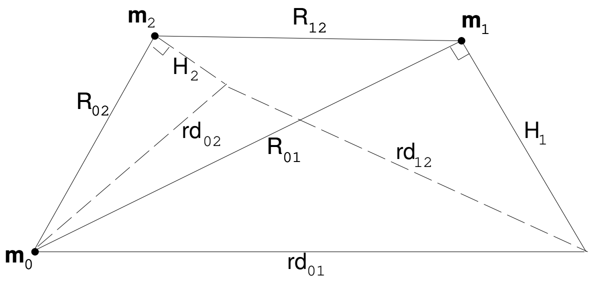

Figure 2-4: Geometrically interpreting the inverted solution for scale

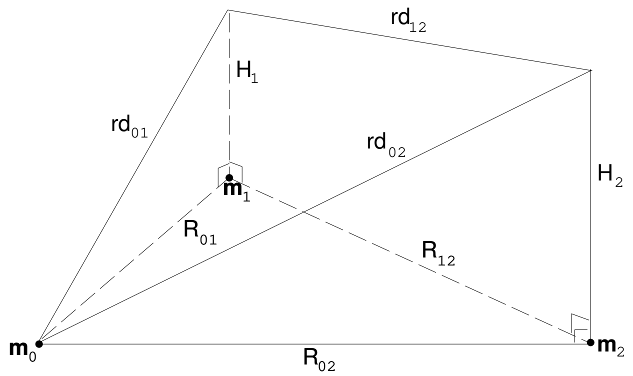

Figure 2-5: Geometrically interpreting the inverted solution for scale

inverted in Equations 2.26-2.28. In effect, what is happening is that instead of scaling down the model triangle and projecting it orthographically onto the image triangle, the image triangle is being scaled up and projected orthographically onto the model triangle, where orthographically means projected along rays that are perpendicular to the model triangle. This means we can rotate the solid in Fig. 2-4 so that the three model points are in the image plane, and, as shown in Fig. 2-5, obtain for the inverted solution a weak-perspective geometry that is analogous to the true geometry.

2.5 Special Configurations of the Points

2.5.1 Model triangle is parallel to the image plane

The two solutions for the scale factor are the same when $b^2 - ac = 0$, and here I demonstrate that geometrically this corresponds to the plane containing the three model points being parallel to the image plane. Before proving this, let us establish the existence of the solution for scale in this special case of $b^2 - ac = 0$. Looking at Equation 2.21,

$$b^2 - ac = 0 \quad \Longrightarrow \quad b \pm \sqrt{b^2 - ac} = b \quad \Longrightarrow \quad s = \sqrt{\frac{b}{a}} \tag{2.30}$$

is a solution to the biquadratic since $a > 0$ and $b > 0$.

Using Equation 2.22, it can be shown that $b^2 - ac = 0$ exactly when $\phi = \pm\psi$ or $\phi = \pm\psi + \pi$ and $\frac{d_{01}}{R_{01}} = \frac{d_{02}}{R_{02}}$ (see Appendix B.3). From this result and Equations 2.18 and 2.20,

$$s \quad = \quad \sqrt{\frac{b}{a}} \quad = \quad \sqrt{\frac{\sqrt{c}}{\sqrt{a}}} \quad = \quad \sqrt{\frac{|d_{01} d_{02} \sin\phi|}{|R_{01} R_{02} \sin\psi|}} \quad = \quad \frac{d_{01}}{R_{01}} = \frac{d_{02}}{R_{02}} \tag{2.31}$$

$$\Longrightarrow \qquad h_1 = \sqrt{(sR_{01})^2 - d_{01}^2} = 0, \qquad h_2 = \sqrt{(sR_{02})^2 - d_{02}^2} = 0.$$

Thus $b^2 - ac = 0$ only if the model triangle is parallel to the image plane. Conversely, if the model triangle is parallel to the image plane, it must be that $\phi = \psi$. Further, in this case $h_1 = h_2 = 0$, so that $s = \frac{d_{01}}{R_{01}} = \frac{d_{02}}{R_{02}}$, which implies that $b^2 - ac = 0$.

Since the two solutions are the same, we know that $s_1 = s_2 = \frac{1}{r}$. Notice in Fig. 2-3 left and Fig. 2-4 that the geometric interpretations for the two solutions for scale collapse to the same solution when $h_1 = h_2 = H_1 = H_2 = 0$ and $s = \frac{1}{r}$. As a result, when there is one solution for scale, there is also one solution for (h_1, h_2) and (H_1, H_2), albeit $(0, 0)$.

2.5.2 Model triangle is perpendicular to the image plane

The situation where the model triangle is perpendicular to the image plane is of interest since the projection is a line. Note, however, that the solution given earlier makes no exception for this case as long as the model triangle is not degenerate. As for what

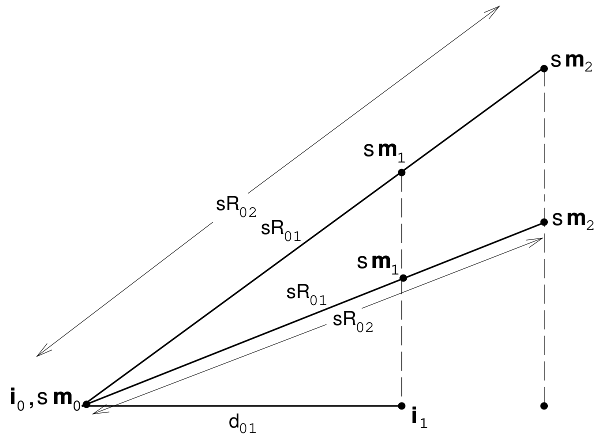

Figure 2-6: Special case where model triangle is a line. The repeated labels correspond to two different solutions for the position of the model that leave sm_1 projecting onto i_1. For both solutions sm_2 projects onto the same image point.

happens in this case, since the image triangle is a line, we know $\psi = 0$ or $\psi = \pi \Longrightarrow c = 0 \Longrightarrow$ Equation 2.21 becomes

$$s = \sqrt{\frac{b \pm \sqrt{b^2}}{a}} = \sqrt{\frac{2b}{a}}\ ,\ 0. \tag{2.32}$$

From Section 2.4, of the two solutions for scale, the true one is $\sqrt{\frac{2b}{a}}$ and the inverted one is 0.

To see why the inverted solution is zero, recall that the solution can be viewed as scaling and projecting the image triangle onto the model triangle, using for scale $r = \frac{1}{s}$, which in this case does not exist. Since the image triangle is a line, graphically this amounts to trying to scale a line so that it can project as a triangle, which is not possible.

2.5.3 Model triangle is a line

This is the one case where the solution for the scale fails, and it fails because a, which is a measure of the area of the model triangle, is zero. Despite this fact, we can determine when a solution exists. First, we know that the image triangle must be a line as well. To see if this condition is enough, consider looking for a 3D rotation and scale that leaves sm_1 orthographically projecting onto i_1 as in Fig. 2-6. Observe that every such rotation and scale leaves sm_2 projecting onto the same point in the image. This means that for a solution to exist, it must be that $\frac{d_{01}}{R_{01}} = \frac{d_{02}}{R_{02}}$. Even when the image triangle is a line, this in general is not true. When it is true, there is an infinity of solutions corresponding to every scaled rotation that leaves sm_1 projecting onto i_1.

Another way to look at this situation is to notice that the model triangle being a line when using the true solution is analogous to the image triangle being a line when using the inverted solution. From Section 2.5.2, the scale factor for the inverted solution does not exist unless $a = 0$, which supports that in this case the scale factor does not exist unless $c = 0$.

2.6 Stability

This section points out two situations to be careful of when using the pose solution. The first is when the model points are nearly collinear, because the solution is near a singularity (Section 2.5.3).

The second situation is when the model triangle is parallel to the image plane. Since the pose solution is unique for any pair of model and image triangles, for each of the three image points there is always some direction in which it can move such that its corresponding model point undergoes 3D rotation without scale (see Fig. 2-7 left). In general, when a model point is rotating in space around a line in the image plane, a movement by its corresponding image point by an amount Δx causes the altitude of the model point to change by an amount Δh (see Fig. 2-7 right), according to

$$x^2 + h^2 = (x - \Delta x)^2 + (h + \Delta h)^2 \tag{2.33}$$

Expanding gives $\Delta h^2 + 2h\Delta h - 2x\Delta x + \Delta x^2 = 0$, which we can solve for Δh:

$$\Delta h = -h \pm \sqrt{h^2 + 2x\Delta x - \Delta x^2} \tag{2.34}$$

When Δx is small and $h = 0$, we have $\Delta h = \pm\sqrt{2x\Delta x}$. Thus, depending on $\sqrt{2x}$, a small change in x could lead to a large change in h, which may cause instability. Note that this instability is inherent in the problem, and so care should be taken when using any solution for 3D pose.

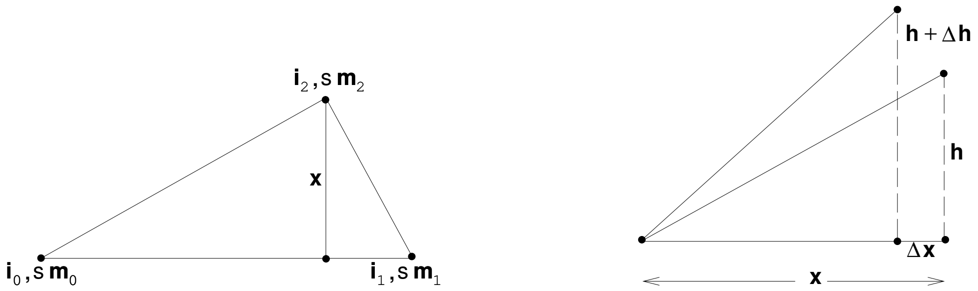

Figure 2-7: Unstable situation. Left: Looking straight down at the image. If $\vec{i}_2$ moves along the perpendicular to the line between $\vec{i}_0$ and $\vec{i}_1$ while the other points remain fixed, then $s\vec{m}_2$ will rotate in space around that line to follow $\vec{i}_2$. Right: Looking from the side. One of the model points is rotating in space around a line in the image plane.

2.7 Derivation of Direct Alignment

To compute the position in the image of a fourth model point, I first use the weak-perspective pose solution to compute its 3D position in camera-centered coordinates. I then project the camera-centered model point under weak-perspective and obtain the image position without having to calculate a model-to-image transformation. Let the image points be $\vec{i}_0 = (x_0, y_0)$, $\vec{i}_1 = (x_1, y_1)$, and $\vec{i}_2 = (x_2, y_2)$. Given s, h_1, h_2, we can invert the projection to get the three model points:

$$\vec{m}_0 = \frac{1}{s}(x_0, y_0, w) \qquad \vec{m}_1 = \frac{1}{s}(x_1, y_1, h_1 + w) \qquad \vec{m}_2 = \frac{1}{s}(x_2, y_2, h_2 + w), \tag{2.35}$$

where w is an unknown offset in a direction normal to the image plane.

Given three noncollinear 2D points, $\vec{q_0}$, $\vec{q_1}$, and $\vec{q_2}$, a fourth 2D point $\vec{q_3}$ can be uniquely represented by its affine coordinates [Graustein30], (α, β), which are given by the equation $\vec{q_3} = \alpha(\vec{q_1} - \vec{q_0}) + \beta(\vec{q_2} - \vec{q_0}) + \vec{q_0}$. Given three noncollinear 3D points, $\vec{p_0}$, $\vec{p_1}$, and $\vec{p_2}$, we can uniquely represent any other 3D point $\vec{p_3}$ in terms of what I shall call its "extended affine coordinates," (α, β, γ):

$$\vec{p_3} = \alpha(\vec{p_1} - \vec{p_0}) + \beta(\vec{p_2} - \vec{p_0}) + \gamma(\vec{p_1} - \vec{p_0}) \times (\vec{p_2} - \vec{p_0}) + \vec{p_0} \tag{2.36}$$

Let (α, β, γ) be the extended affine coordinates of the fourth model point in terms of the matched three, which I assume are noncollinear. Let $x_{01} = x_1 - x_0$, $y_{01} = y_1 - y_0$, $x_{02} = x_2 - x_0$, and $y_{02} = y_2 - y_0$. Then, using the three camera-centered model points such that $\vec{p_0} = \vec{m_0}$, $\vec{p_1} = \vec{m_1}$, and $\vec{p_2} = \vec{m_2}$,

$$\begin{aligned} \vec{p_1} - \vec{p_0} &= \frac{1}{s}(x_{01}, y_{01}, h_1) && (2.37) \\ \vec{p_2} - \vec{p_0} &= \frac{1}{s}(x_{02}, y_{02}, h_2) && (2.38) \\ (\vec{p_1} - \vec{p_0}) \times (\vec{p_2} - \vec{p_0}) &= \frac{1}{s^2}(y_{01}h_2 - y_{02}h_1, x_{02}h_1 - x_{01}h_2, x_{01}y_{02} - x_{02}y_{01}) \,. && (2.39) \end{aligned}$$

Next, substitute Equations 2.37-2.39 into Equation 2.36 to get the 3D location of the fourth point:

$$\begin{aligned} \vec{m_3} &= \frac{1}{s}\alpha(x_{01}, y_{01}, h_1) + \frac{1}{s}\beta(x_{02}, y_{02}, h_2) \\ &\quad + \gamma\frac{1}{s^2}(y_{01}h_2 - y_{02}h_1, -x_{01}h_2 + x_{02}h_1, x_{01}y_{02} - x_{02}y_{01}) + \frac{1}{s}(x_0, y_0, w) \\ &= \frac{1}{s}(\alpha x_{01} + \beta x_{02} + \gamma\frac{y_{01}h_2 - y_{02}h_1}{s} + x_0, \\ &\quad \alpha y_{01} + \beta y_{02} + \gamma\frac{-x_{01}h_2 + x_{02}h_1}{s} + y_0, \\ &\quad \alpha h_1 + \beta h_2 + \gamma\frac{x_{01}y_{02} - x_{02}y_{01}}{s} + w) && (2.40) \end{aligned}$$

Let Π represent an orthogonal projection along the z axis. To project, multiply by the scale factor s and drop the z coordinate:

$$\begin{aligned} \Pi(s\vec{m_3}) &= (\alpha x_{01} + \beta x_{02} + \gamma(y_{01}H_2 - y_{02}H_1) + x_0, \\ &\quad \alpha y_{01} + \beta y_{02} + \gamma(-x_{01}H_2 + x_{02}H_1) + y_0) && (2.41) \end{aligned}$$

Notice that the unknown offset w has dropped out. This expression computes the image position of $\vec{p_3}$ from its extended affine coordinates, from the image points, and from H_1 and H_2, the altitudes in the weak-perspective geometry. It should be kept in mind that the altitudes H_1 and H_2 depend on the specific imaging geometry; that is, they depend on the pose of the model.

Equation 2.35 gives the three matched model points in camara-centered coordinates without having to compute a rigid 3D transformation. This should reduce the cost of computing the camera-centered locations of these points, which will speed-up recognition systems to the extent this computation is important. Further advantages of the direct method are that it is more intuitive because it is more directly connected to the geometry, and it may be simpler to use.

It may be worthwhile to observe that Equation 2.41, the expression for the fourth point, can be rewritten as a weighted sum of the three image points:

$$\begin{aligned}
\Pi(s\vec{m}_3) &= (\alpha x_{01} + \beta x_{02} + \gamma(y_{01}H_2 - y_{02}H_1) + x_0, \\
&\quad \alpha y_{01} + \beta y_{02} + \gamma(-x_{01}H_2 + x_{02}H_1) + y_0) \\
&= (\alpha x_1 + \gamma H_2 y_1, \alpha y_1 - \gamma H_2 x_1) - (\alpha x_0 + \gamma H_2 y_0, \alpha y_0 - \gamma H_2 x_0) + \\
&\quad (\beta x_2 - \gamma H_1 y_2, \beta y_2 + \gamma H_1 x_2) - (\beta x_0 - \gamma H_1 y_0, \beta y_0 + \gamma H_1 x_0) + \\
&\quad (x_0, y_0) \\
&= \begin{bmatrix} 1-\alpha-\beta & \gamma(H_1 - H_2) \\ -\gamma(H_1 - H_2) & 1-\alpha-\beta \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \\
&\quad \begin{bmatrix} \alpha & \gamma H_2 \\ -\gamma H_2 & \alpha \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} \beta & -\gamma H_1 \\ \gamma H_1 & \beta \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}
\end{aligned}$$

Let $\mathbf{R}_\theta$ represent a 2D rotation matrix that rotates by an angle θ. Then

$$\Pi(s\vec{m}_3) = \delta_0 \mathbf{R}_{\theta_0} \vec{i}_0 + \delta_1 \mathbf{R}_{\theta_1} \vec{i}_1 + \delta_2 \mathbf{R}_{\theta_2} \vec{i}_2, \tag{2.42}$$

where

$$\delta_0 = \sqrt{(1-\alpha-\beta)^2 + (\gamma(H_1 - H_2))^2} \tag{2.43}$$

$$\delta_1 = \sqrt{\alpha^2 + (\gamma H_2)^2} \tag{2.44}$$

$$\delta_2 = \sqrt{\beta^2 + (\gamma H_1)^2} \tag{2.45}$$

$$\begin{array}{ll} \cos\theta_0 = \frac{1-\alpha-\beta}{\delta_0} & \sin\theta_0 = \frac{-\gamma(H_1 - H_2)}{\delta_0} \\ \cos\theta_1 = \frac{\alpha}{\delta_1} & \sin\theta_1 = \frac{-\gamma H_2}{\delta_1} \\ \cos\theta_2 = \frac{\beta}{\delta_2} & \sin\theta_2 = \frac{\gamma H_1}{\delta_2} \end{array} \tag{2.46}$$

Thus, we can view the computation as a 2D rotation and scale of each image point separately followed by a sum of the three. It is important to keep in mind, however, that the rotations and scales themselves depend on the image points, because of H_1 and H_2.

When the model is planar, the form of Equation 2.42 facilitates understanding the effects of error in the image points. Error in the locations of the matched image points leads to uncertainty in the image location of the fourth model point. Suppose that the true locations of the matched image points are known to be within a few, say ϵ_i, pixels of their nominal locations, for $i = 0, 1, 2$. Let $\vec{i_i}$ and $\vec{c_i}$ be the true and nominal locations of an image point, for $i = 0, 1, 2$. Then, for some $\vec{e_0}$, $\vec{i_0} = \vec{c_0} + \vec{e_0}$, where $\| \vec{e_0} \| = \epsilon_0$, and similarly for $\vec{i_1}$ and $\vec{i_2}$. Then

$$\begin{aligned} \Pi(s\vec{m_3}) &= \delta_0 \mathbf{R}_{\theta_0} \vec{i_0} + \delta_1 \mathbf{R}_{\theta_1} \vec{i_1} + \delta_2 \mathbf{R}_{\theta_2} \vec{i_2} \\ &= (\delta_0 \mathbf{R}_{\theta_0} \vec{c_0} + \delta_1 \mathbf{R}_{\theta_1} \vec{c_1} + \delta_2 \mathbf{R}_{\theta_2} \vec{c_2}) + (\delta_0 \mathbf{R}_{\theta_0} \vec{e_0} + \delta_1 \mathbf{R}_{\theta_1} \vec{e_1} + \delta_2 \mathbf{R}_{\theta_2} \vec{e_2}) \end{aligned}$$

When the fourth point is in the plane of the first three, $\gamma = 0$, so that the scales, δ_0, δ_1, and δ_2, and 2D rotations, $\mathbf{R}_{\theta_0}$, $\mathbf{R}_{\theta_1}$, and $\mathbf{R}_{\theta_2}$, are all constant (see Equations 2.43-2.46). This means that the first term in parentheses is just the nominal image location of the fourth model point. Since $\vec{e_0}$, $\vec{e_1}$, and $\vec{e_2}$ move around circles, the 2D rotations in the second term can be ignored. Further, since these error vectors move independently around their error circles, their radii simply sum together. Therefore, the region of possible locations of the fourth model point is bounded by a circle of radius $\delta_0\epsilon_0 + \delta_1\epsilon_1 + \delta_2\epsilon_2$ that is centered at the nominal point. By plugging $\gamma = 0$ into Equations 2.43-2.45, we get that

$$\delta_0 = |1 - \alpha - \beta|\,, \qquad \delta_1 = |\alpha|\,, \qquad \delta_2 = |\beta|\,,$$

Assuming $\epsilon_0 = \epsilon_1 = \epsilon_2 = \epsilon$, this implies that the uncertainty in the image location of a fourth point is bounded by a circle with radius $(|1 - \alpha - \beta| + |\alpha| + |\beta|)\epsilon$ and with its center at the nominal point, which repeats the result given earlier by Jacobs [Jacobs91].

Although the non-planar case clearly is more complicated, since the scales and 2D rotations are no longer constant, Equation 2.42 may prove useful for obtaining bounds on the effects of error in this situation as well.

2.8 Review of Previous Solutions

There have been several earlier solutions to the weak-perspective three-point problem, notably by Kanade and Kender [Kanade83], Cyganski and Orr [Cyganski85] [Cyganski88], Ullman [Ullman86] [Huttenlocher87], Huttenlocher and Ullman [Huttenlocher88]

[Huttenlocher90] [Ullman89], and Grimson, Huttenlocher, and Alter [Grimson92a]. All the previous solutions compute the 3D pose by going through a 3D rigid transformation or a 2D affine transformation relating the model to the image. A 2D affine transform is a linear transform plus a translation, and it can be applied to any object lying in the plane. All but Ullman's and Grimson, Huttenlocher, and Alter's solutions compute an affine transformation between the three model and image points. Also, all but Kanade and Kender's solution compute a model-to-image rigid transformation, either via a rotation matrix or via Euler angles.

Not all of the solutions directly solve the weak-perspective three-point problem. The earliest solution, which was given by Kanade and Kender in 1983, applies Kanade's skewed-symmetry constraint to recover the 3D orientation of a symmetric, planar pattern [Kanade83]. More precisely, Kanade and Kender showed how to compute the 3D orientation of the plane containing a symmetric, planar pattern from a 2D affine transform between an image of the pattern and the pattern itself. To apply this result to the weak-perspective three-point problem, the three points can be used to construct a symmetric, planar pattern, and a 2D affine transform can be computed from two sets of three corresponding points. The solution was shown to exist and to give two solutions related by a reflective ambiguity, assuming that the determinant of the affine transform is positive.

The remaining methods all concentrate on computing the 3D rigid transform from the model to the image. In 1985, while presenting a system for recognizing planar objects, Cyganksi and Orr showed how to use higher-order moments to compute a 2D affine transform between planar regions [Cyganski85] [CyganskiOrr88]. Given the affine transform, they listed expressions for computing the 3D Euler angles from the 2D affine transform[1]. They did not, however, discuss how they derived the expressions.

The next method is the solution given by Ullman in 1986 [Ullman86], which appeared again in [Huttenlocher87]. The paper included a proof that the solution for the scale factor is unique and the solution for the rotation matrix is unique up to an inherent two-way ambiguity. (This corresponds to the ambiguity in H_1 and H_2.) Yet Ullman did not show the solution exists. When it does exist, Ullman described a method for obtaining the rotation matrix and scale factor.

In 1988, Huttenlocher and Ullman gave another solution, and, in the process, gave

[1]The expressions that appear in [Cyganski85] contain typesetting errors, but are listed correctly in [Cyganski88].

the first complete proof that the solution both exists and is unique (up to the two-way ambiguity) [Huttenlocher88] [Huttenlocher90] [Ullman89]. Like Kanade and Kender, and Cyganski and Orr, Huttenlocher and Ullman's solution relies on a 2D affine transform. The solution itself is based on algebraic constraints derived from rigidity, which are used to recover the elements of the scaled rotation matrix.

The last solution, which was published this year, was developed by Grimson, Huttenlocher, and Alter for the purpose of analyzing the effects of image noise on error in transformation space [Grimson92a]. Towards this end, the method facilitates computing how a small perturbation in each transformation parameter propagates to uncertainty ranges in the other parameters.

2.9 Presentation of Three Previous Solutions

The solutions discussed in the previous section differ significantly in how they compute the transformation, and, as a result, each one can provide different insights into solving related problems, such as error analysis in alignment-based recognition and pose clustering. It seems useful, then, to present the previous solutions in detail, so they conveniently can be referred to and compared.

The first method presented is Ullman's solution, which the first part of this chapter extended. After that, I give Huttenlocher and Ullman's solution. Lastly, I present the method of Grimson, Huttenlocher, and Alter. I do not present Kanade and Kender's method nor Cyganski and Orr's, because Kanade and Kender did not directly solve the weak-perspective three-point problem, and Cyganski and Orr did not detail their solution.

It should be pointed out that the presentations here differ somewhat from the ones given by the original authors, but the ideas are the same. Basically, the presentations emphasize the steps that recover the 3D pose while being complete and concise.

In the following presentations, we are looking for a rigid transform plus scale that aligns the model points to the image points. In all methods, we are free to move rigidly the three image points or the three model points wherever we wish, since this amounts to tacking on an additional transform before or after the aligning one. For example, this justifies the assumption made below that the plane of the model points is parallel to the image plane.

For consistency, the same notation as in Sections 2.2 and 2.4 is used in the proofs that follow: Let the model points be $\vec{m}_0$, $\vec{m}_1$, $\vec{m}_2$ and the image points be $\vec{i}_0$, $\vec{i}_1$, $\vec{i}_2$, with the respective distances between the points being R_{01}, R_{02}, and R_{12} for the model points, and d_{01}, d_{02}, and d_{12} for the image points.

2.9.1 Overview

Initially, all three methods compute a transformation that brings the model into image coordinates, such that the plane of the three matched model points is parallel to the image plane and such that $\vec{m}_0$ projects onto $\vec{i}_0$, which has been translated to the origin. The three methods then compute the out-of-plane rotation and scale that align the matched model and image points. In so doing, the methods all end up solving a biquadratic equation.

In Ullman's method, the model and image points are further transformed via rotations around the z axis to align $\vec{m}_1$ and $\vec{i}_1$ along the x axis. Then the 3D rotation matrix for rotating successively around the x and y axes is expressed in terms of Euler angles. This leads to a series of three equations in three unknowns, which are solved to get a biquadratic in the scale factor. To get the elements of the rotation matrix, the solution for scale factor is substituted back into the original three equations.

Instead of further rotating the model and image points, Huttenlocher and Ullman compute an affine transform between them, which immediately gives the top-left submatrix of the scaled rotation matrix. Then by studying what happens to two equal-length vectors in the plane, a biquadratic is obtained. The scale factor and the remaining elements of the scaled rotation matrix are found using the algebraic constraints on the columns of a scaled rotation matrix.

Like Ullman did, Grimson, Huttenlocher, and Alter rotate the model further to align $\vec{m}_1$ and $\vec{i}_1$. The desired out-of-plane rotation is expressed in terms of two angles that give the rotation about two perpendicular axes in the plane. Next, Rodrigues' formula, which computes the 3D rotation of a point about some axis, is used to eliminate the scale factor and obtain two constraints on the two rotation angles. The two constraints are solved to get a biquadratic in the cosine of one of the angles. Its solution is substituted back to get the other angle and the scale factor, which can be used directly by Rodrigues' formula to transform any other model point.

As mentioned in the introduction, Ullman's solution is incomplete because it does not show which of the two solutions for the scale factor is correct; actually, the solution is completed by the result given in Section 2.4.1 of this chapter. Similar to Ullman's method, Grimson, Huttenlocher, and Alter's solution has the same drawback of not showing which solution to its biquadratic is correct. Huttenlocher and Ullman, on the other hand, have no such problem because it turns out that one of the two solutions to their biquadratic is obviously not real, and so it immediately is discarded.

2.9.2 Ullman's method

This section gives Ullman's solution to the weak-perspective three-point problem. The main idea is first to transform the three model points to the image plane and then solve for the scale and out-of-plane rotation that align the transformed points.

Specifically, the model points first are rigidly transformed to put the three model points in the image plane with $\vec{m}_0$ at the origin of the image coordinate system and $\vec{m}_1 - \vec{m}_0$ aligned with the x axis. After rigidly transforming the model points, the resulting points can be represented by $(0,0,0)$, $(\bar{x_1},0,0)$, and $(\bar{x_2},\bar{y_2},0)$. Similarly, let the image points be rigid transformed to put $\vec{i}_0$ at the origin and $\vec{i}_1 - \vec{i}_0$ along the x axis, and let the resulting image points be $(0,0,0)$, $(x_1,0,0)$, and $(x_2,y_2,0)$.

Next, we break the out-of-plane rotation into a rotation around the x axis by an angle θ followed by a rotation around the y axis by an angle ϕ, as pictured in Fig. 2-8. The corresponding rotation matrix is

$$\begin{aligned}\mathbf{R} &= \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \\ &= \begin{bmatrix} \cos\phi & \sin\phi\sin\theta & \sin\phi\cos\theta \\ 0 & \cos\theta & -\sin\theta \\ -\sin\phi & \cos\phi\sin\theta & \cos\phi\cos\theta \end{bmatrix} \end{aligned} \tag{2.47}$$

After rotation and scale, $(0,0,0)$, $(\bar{x_1},0,0)$, and $(\bar{x_2},\bar{y_2},0)$ become $(0,0,0)$, $(x_1,0,z_1)$, and (x_2,y_2,z_2), respectively, where z_1 and z_2 are unknown. Thus, we need to find θ, ϕ, and s such that

$$\begin{aligned} s\mathbf{R}(\bar{x_1},0,0) &= (x_1,0,z_1) \\ s\mathbf{R}(\bar{x_2},\bar{y_2},0) &= (x_2,y_2,z_2) \end{aligned}$$

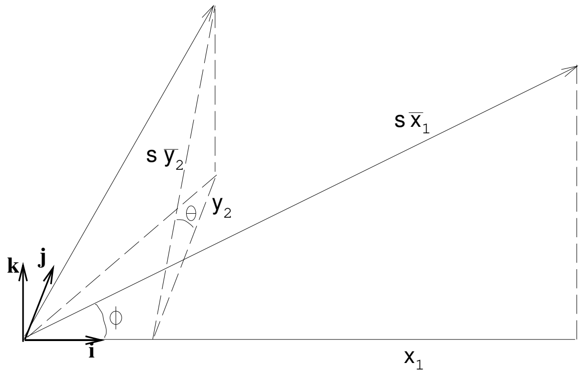

Figure 2-8: Interpreting the out-of-plane rotation angles in Ullman's method.

Expanding the first two rows of **R** yields three equations in three unknowns:

$$s\bar{x}_1 \cos\phi = x_1 \tag{2.48}$$
$$s\bar{y}_2 \cos\theta = y_2 \tag{2.49}$$
$$s\bar{x}_2 \cos\phi - s\bar{y}_2 \sin\phi \sin\theta = x_2 \tag{2.50}$$

Fig. 2-8 gives a graphical interpretation of the first two equations. Substituting Equations 2.48 and 2.49 along with expressions for $\sin\phi$ and $\sin\theta$ into Equation 2.50 yields a biquadratic in the scale factor s:

$$as^4 - bs^2 + c = 0, \tag{2.51}$$

where

$$a = \bar{x}_1{}^2 \bar{y}_2{}^2 \tag{2.52}$$
$$b = x_1^2(\bar{x}_2{}^2 + \bar{y}_2{}^2) + \bar{x}_1{}^2(x_2^2 + y_2^2) - 2x_1 x_2 \bar{x}_1 \bar{x}_2 \tag{2.53}$$
$$c = x_1^2 y_2^2 \tag{2.54}$$

The positive solutions for s are given by

$$s = \sqrt{\frac{b \pm \sqrt{b^2 - 4ac}}{2a}} \tag{2.55}$$

In general there can be one, two, or no solutions for s. Ullman makes no further attempt to determine when or if each solution arises, except to refer to a uniqueness proof he gives earlier in the paper. The uniqueness proof implies there can be at most one solution for s, but does not say which solution it is or whether it can be either one at different times.

Given s, the rotation matrix $\mathbf{R}$ is obtained using $\cos\phi = \frac{x_1}{s\bar{x_1}}$ and $\cos\theta = \frac{y_2}{s\bar{y_2}}$ in Equation 2.47. One difficulty with this is that we do not know the signs of $\sin\theta$ and $\sin\phi$; this leaves four possibilities for the pair $(\sin\theta, \sin\phi)$. In his uniqueness proof, Ullman points out that the inherent reflective ambiguity corresponds to multiplying simultaneously the elements r_{13}, r_{23}, r_{31}, and r_{32} of $\mathbf{R}$ by -1. In Equation 2.47, the signs of those elements also are inverted when both $\sin\theta$ and $\sin\phi$ are multiplied by -1, which, visually, corresponds to reflecting the model points about the image plane (Fig. 2-8). Still, we have no way to know which of the two pairs of solutions is correct. One way to proceed is to try both and see which solution pair aligns the points.

2.9.3 Huttenlocher and Ullman's method

First, assume the plane containing the model points is parallel to the image plane. Then subtract out $\vec{m}_0$ and $\vec{i}_0$ from the model and image points, respectively, to align them at the origin. Let the resulting model points be $(0,0,0)$, $(\bar{x_1}, \bar{y_1}, 0)$, and $(\bar{x_2}, \bar{y_2}, 0)$, and the resulting image points be $(0,0)$, (x_1, y_1), and (x_2, y_2). At this point, what is left is to compute the scaled rotation matrix that brings $(\bar{x_1}, \bar{y_1}, 0)$ and $(\bar{x_2}, \bar{y_2}, 0)$ to (x_1, y_1, z_1) and (x_2, y_2, z_2), respectively, where z_1 and z_2 are unknown. That is, we need

$$\begin{aligned} s\mathbf{R}(\bar{x_1}, \bar{y_1}, 0) &= (x_1, y_1, z_1) \\ s\mathbf{R}(\bar{x_2}, \bar{y_2}, 0) &= (x_2, y_2, z_2). \end{aligned}$$

Letting $l_{11} = sr_{11}$, $l_{12} = sr_{12}$, etc., and focusing on the first two rows of the rotation matrix, we get two sets of equations:

$$l_{11}\bar{x_1} + l_{12}\bar{y_1} = x_1 \tag{2.56}$$
$$l_{11}\bar{x_2} + l_{12}\bar{y_2} = x_2 \tag{2.57}$$

$$l_{21}\bar{x_1} + l_{22}\bar{y_1} = y_1 \tag{2.58}$$
$$l_{21}\bar{x_2} + l_{22}\bar{y_2} = y_2, \tag{2.59}$$

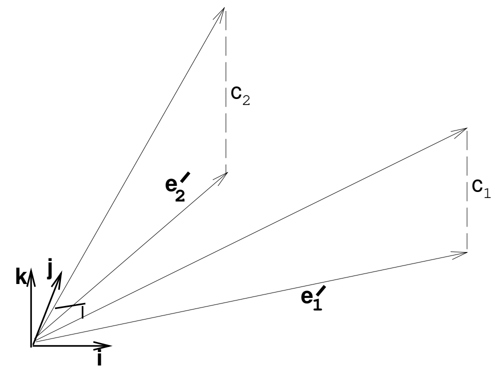

Figure 2-9: Projecting two orthogonal same-length vectors in Huttenlocher and Ullman's method.

which give $\begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{bmatrix}$, the top-left sub-matrix of the scaled rotation matrix. Note that this step fails if the determinent, $\bar{x}_1\bar{y}_2 - \bar{x}_2\bar{y}_1$, equals zero.

Next, we make a digression to consider what happens to two orthogonal, equal-length vectors in the plane, $\vec{e_1}$ and $\vec{e_2}$. Since $\vec{e_1}$ and $\vec{e_2}$ are in the plane, we can apply the sub-matrix just computed to obtain the resulting vectors, $\vec{e_1}'$ and $\vec{e_2}'$:

$$\vec{e_1}' = \begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{bmatrix} \vec{e_1}, \qquad \vec{e_2}' = \begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{bmatrix} \vec{e_2} \tag{2.60}$$

When a model is transformed, $\vec{e_1}$ and $\vec{e_2}$ undergo a rigid transformation plus scale before projection. As shown in Fig. 2-9, after transformation these vectors become $\vec{e_1}' + c_1\hat{z}$ and $\vec{e_2}' + c_2\hat{z}$. Since a scaled, rigid transform preserves angles and ratios of lengths between vectors, and since $\vec{e_1} \cdot \vec{e_2} = 0$ and $\| \vec{e_1} \| = \| \vec{e_2} \|$, it must be that

$$(\vec{e_1}' + c_1\hat{z}) \cdot (\vec{e_2}' + c_2\hat{z}) = 0$$

$$\| \vec{e_1}' \| + c_1^2 = \| \vec{e_2}' \| + c_2^2.$$

These two equations simplify to

$$\begin{aligned} c_1 c_2 &= k_1 \\ c_1^2 - c_2^2 &= k_2 \end{aligned}$$

where

$$\begin{aligned} k_1 &= -\vec{e_1}' \cdot \vec{e_2}' & (2.61) \\ k_2 &= \| \vec{e_2}' \| - \| \vec{e_1}' \| & (2.62) \end{aligned}$$

Substituting for $c_2 = \frac{k_1}{c_1}$ in the second equation leads to a biquadratic in c_1:

$$c_1^4 - k_2 c_1^2 - k_1^2 = 0 \tag{2.63}$$

The general solution is

$$c_1 = \pm\sqrt{\frac{1}{2}\left(k_2 \pm \sqrt{k_2^2 + 4k_1^2}\right)} \, .$$

Conveniently, the inner discriminant always is greater than or equal to zero. Furthermore, since $4k_1^2 \geq 0$, the real solutions are given by

$$c_1 = \pm\sqrt{\frac{1}{2}\left(k_2 + \sqrt{k_2^2 + 4k_1^2}\right)} \, , \tag{2.64}$$

since otherwise the outer discriminant is less than zero.

These two solutions for c_1 give two corresponding solutions for c_2, which from Fig. 2-9 can be seen to correspond to a reflection about the image plane.

The solution for c_2 does not work when $c_1 = 0$. In this case,

$$c_2 = \pm\sqrt{-k_2} = \pm\sqrt{\| \vec{e_1}' \| - \| \vec{e_2}' \|} \, . \tag{2.65}$$

This gives two solutions for c_2, if it exists, which can be seen as follows. Since $c_1 = 0$, $\vec{e_1}$ ends up in the plane, so that that the length of $\vec{e_1}$ is just scaled down by s, whereas the length of $\vec{e_2}$ reduces both by being scaled down and by projection. Consequently, $\| \vec{e_2}' \| \leq \| \vec{e_1}' \|$, and, therefore, c_2 exists.

Given c_1 and c_2, we can recover two more elements of the scaled rotation matrix. Since $\vec{e_1}$ and $\vec{e_2}$ are in the plane, we know that $s\mathbf{R}\vec{e_1} = \vec{e_1}' + c_1\hat{z}$ and $s\mathbf{R}\vec{e_2} = \vec{e_2}' + c_2\hat{z}$. Focusing on the last row of the scaled rotation matrix, we get the two equations $l_{31} = c_1$ and $l_{32} = c_2$.

At this point, we have the first two columns of $s\mathbf{R}$, and, from the constraints on the columns of a rotation matrix, we can get the last column from the cross product of the first two. In total, this gives

$$s\mathbf{R} = \begin{bmatrix} l_{11} & l_{12} & \frac{1}{s}(c_2 l_{21} - c_1 l_{22}) \\ l_{21} & l_{22} & \frac{1}{s}(c_1 l_{12} - c_2 l_{11}) \\ c_1 & c_2 & \frac{1}{s}(l_{11} l_{22} - l_{12} l_{21}) \end{bmatrix} \tag{2.66}$$

Since the columns of a rotation matrix have unit length, we know

$$s = \sqrt{l_{11}^2 + l_{21}^2 + c_1^2} = \sqrt{l_{12}^2 + l_{22}^2 + c_2^2}\,. \tag{2.67}$$

Notice that the ambiguity in c_1 and c_2 inverts the signs of the appropriate elements of the rotation matrix as discussed in Section 2.9.2.

2.9.4 Grimson, Huttenlocher, and Alter's method

Grimson et al. gave another solution to the weak-perspective three point problem in order to get a handle on how small perturbations affect the individual transformation parameters.

To start, assume the plane containing the model points is parallel to the image plane. Next, rigidly transform the model points so that $\vec{m}_0$ projects to $\vec{i}_0$ and $\vec{m}_1 - \vec{m}_0$ projects along $\vec{i}_1 - \vec{i}_0$. Let Π represent an orthogonal projection along the z axis, and in general let $\vec{v}^{\perp}$ be the 2D vector rotated ninety degrees clockwise from the 2D vector $\vec{v}$. Then the translation is $\vec{i}_0 - \Pi\vec{m}_0$, and the rotation is about $\hat{z}$ by an angle ψ given by

$$\cos\psi = \widehat{m}_{01} \cdot \hat{i}_{01}, \qquad \sin\psi = -\widehat{m}_{01} \cdot \hat{i}_{01}^{\perp}.$$

(see Fig. 2-10).

At this point, assign $\vec{m}_{01} = \vec{m}_1 - \vec{m}_0$, $\vec{m}_{02} = \vec{m}_2 - \vec{m}_0$, $\vec{i}_{01} = \vec{i}_1 - \vec{i}_0$, and $\vec{i}_{02} = \vec{m}_2 - \vec{m}_0$. Also, consider the out-of-plane rotation to be a rotation about $\hat{i}_{01}$ by some angle θ followed by a rotation about $\hat{i}_{01}^{\perp}$ by some angle ϕ. Let us compute where the vectors $\hat{i}_{01}$ and $\hat{i}_{01}^{\perp}$ project to after the two rotations and scale. To do this, we use Rodrigues' formula: Let $\mathbf{R}_{\hat{v},\tau}\vec{p}$ represent a rotation of a point $\vec{p}$ about a direction $\hat{v}$ by an angle τ. Rodrigues' formula is

$$\mathbf{R}_{\hat{v},\tau}\vec{p} = \cos\tau\vec{p} + (1 - \cos\tau)(\hat{v}\cdot\vec{p})\hat{v} + \sin\tau(\hat{v} \times \vec{p}). \tag{2.68}$$

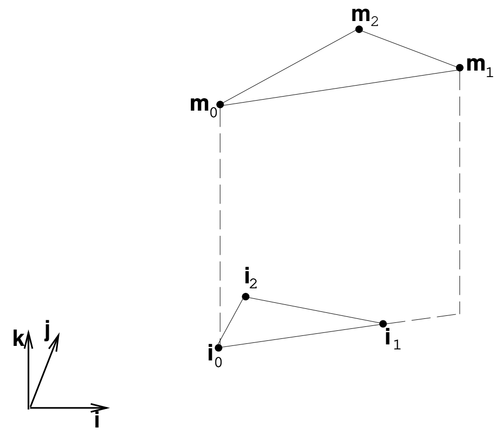

Figure 2-10: After the rotation by ψ in Grimson, Huttenlocher, and Alter's method, the plane of the model points is parallel to the image plane, $\vec{m}_0$ projects onto $\vec{i}_0$, and $\vec{m}_1 - \vec{m}_0$ projects along $\vec{i}_1 - \vec{i}_0$.

Using the formula, we can compute

$$\mathbf{R}_{\widehat{i}_{01}^{\perp},\phi}\mathbf{R}_{\widehat{i}_{01},\theta}\widehat{i}_{01} = \cos\phi\widehat{i}_{01} - \sin\phi\widehat{z} \tag{2.69}$$

$$\mathbf{R}_{\widehat{i}_{01}^{\perp},\phi}\mathbf{R}_{\widehat{i}_{01},\theta}\widehat{i}_{01}^{\perp} = \sin\theta\sin\phi\widehat{i}_{01} + \cos\theta\widehat{i}_{01}^{\perp} + \sin\theta\cos\phi\widehat{z}.$$

Initially, $\vec{m}_{01}$ was rotated about $\widehat{z}$ to align it with $\vec{i}_{01}$. In order for the scaled orthographic projection of $\vec{m}_{01}$ to align with $\vec{i}_{01}$, Equation 2.69 implies that

$$\begin{aligned} s &= \frac{\| \vec{i}_{01} \|}{\| \vec{m}_{01} \|}\frac{1}{\cos\phi} \\ &= \frac{d_{01}}{R_{01}}\frac{1}{\cos\phi}. \end{aligned} \tag{2.70}$$

Then

$$s\Pi\mathbf{R}_{\widehat{i}_{01}^{\perp},\phi}\mathbf{R}_{\widehat{i}_{01},\theta}\widehat{i}_{01} = \frac{d_{01}}{R_{01}}\widehat{i}_{01} \tag{2.71}$$

$$s\Pi\mathbf{R}_{\widehat{i}_{01}^{\perp},\phi}\mathbf{R}_{\widehat{i}_{01},\theta}\widehat{i}_{01}^{\perp} = \frac{d_{01}}{R_{01}}\frac{1}{\cos\phi}(\sin\theta\sin\phi\widehat{i}_{01} + \cos\theta\widehat{i}_{01}^{\perp}) \tag{2.72}$$

Next, we use the expressions in Equations 2.71 and 2.72 to constrain θ and ϕ such that $\vec{m}_{02}$ projects along $\vec{i}_{02}$. When we aligned $\vec{m}_{01}$ and $\vec{i}_{01}$, $\vec{m}_{02}$ rotated to $\mathbf{R}_{\hat{z},\psi}\vec{m}_{02}$. Since $\vec{m}_{02}$ has no $\hat{z}$ component (by assumption), we can represent $\mathbf{R}_{\hat{z},\psi}\vec{m}_{02}$ by

$$R_{02}\cos\xi\hat{i}_{01} + R_{02}\sin\xi\hat{i}_{01}^{\perp},$$

where ξ is a known angle. Consequently, the transformed, projected, and scaled $\vec{m}_{02}$, which must equal $\vec{i}_{02}$, is

$$\begin{aligned}
&s\Pi\mathbf{R}_{\hat{i}_{01}^{\perp},\phi}\mathbf{R}_{\hat{i}_{01},\theta}(R_{02}\cos\xi\hat{i}_{01} + R_{02}\sin\xi\hat{i}_{01}^{\perp}) \\
&= R_{02}\cos\xi(s\Pi\mathbf{R}_{\hat{i}_{01}^{\perp},\phi}\mathbf{R}_{\hat{i}_{01},\theta}\hat{i}_{01}) + R_{02}\sin\xi(s\Pi\mathbf{R}_{\hat{i}_{01}^{\perp},\phi}\mathbf{R}_{\hat{i}_{01},\theta}\hat{i}_{01}^{\perp}) \\
&= R_{02}\cos\xi\left(\frac{d_{01}}{R_{01}}\hat{i}_{01}\right) + R_{02}\sin\xi\left(\frac{d_{01}}{R_{01}}\frac{1}{\cos\phi}(\sin\theta\sin\phi\hat{i}_{01} + \cos\theta\hat{i}_{01}^{\perp})\right) \\
&= \frac{d_{01}}{\cos\phi}\frac{R_{02}}{R_{01}}(\cos\xi\cos\phi + \sin\xi\sin\phi\sin\theta)\hat{i}_{01} + \frac{d_{01}}{\cos\phi}\frac{R_{02}}{R_{01}}(\sin\xi\cos\theta)\hat{i}_{01}^{\perp}.
\end{aligned}$$

Similar to $\mathbf{R}_{\hat{z},\psi}\vec{m}_{02}$, we can represent $\vec{i}_{02}$ as

$$\vec{i}_{02} = d_{02}\cos\omega\hat{i}_{01} + d_{02}\sin\omega\hat{i}_{01}^{\perp},$$

where ω is known. By equating terms we get

$$\frac{d_{01}}{d_{02}}\frac{R_{02}}{R_{01}}(\cos\xi\cos\phi + \sin\xi\sin\phi\sin\theta) = \cos\phi\cos\omega \tag{2.73}$$

$$\frac{d_{01}}{d_{02}}\frac{R_{02}}{R_{01}}(\sin\xi\cos\theta) = \cos\phi\sin\omega. \tag{2.74}$$

These two equations can be solved to get a biquadratic in $\cos\phi$:

$$\sin^2\omega\cos^4\phi - (t^2 + 1 - 2t\cos\omega\cos\xi)\cos^2\phi + t^2\sin^2\xi = 0, \tag{2.75}$$

where

$$t = \frac{R_{02}d_{01}}{R_{01}d_{02}}. \tag{2.76}$$

Since $\mathbf{R}_{\hat{z},\psi}\vec{m}_{01}$ is aligned with $\vec{i}_{01}$, we need $\cos\phi$ to be positive so that $\vec{m}_{01}$ projects in the same direction as $\vec{i}_{01}$. The positive solutions are given by

$$\cos\phi = \frac{1}{|\sin\omega|}\sqrt{\nu \pm \sqrt{\nu^2 - t^2\sin^2\omega\sin^2\xi}} \tag{2.77}$$

with

$$\nu = \frac{1}{2}(1 + t^2 - 2t\cos\omega\cos\xi).$$

This equation gives up to two solutions, but Grimson et al. make no further attempt to show which solutions exists when, except to say the equation gives real solutions only if $\nu \geq 0$ or

$$\cos\omega\cos\xi \leq \frac{1+t^2}{2t}. \tag{2.78}$$

Given ϕ, Equations 2.73 and 2.74 provide θ:

$$\cos\theta = \frac{\sin\omega\cos\phi}{t\sin\xi} \tag{2.79}$$

$$\sin\theta = \frac{\cos\phi(\cos\omega - t\cos\xi)}{t\sin\xi\sin\phi} \tag{2.80}$$

Given any model point $\vec{m}$, we can use the computed angles along with Rodrigues' formula to find its image location. In particular, once $\vec{m}_0$ and $\vec{i}_0$ have been subtracted out, only the scale and 3D rotation are left. The scale is given by Equation 2.70, and, as shown above, the rotation is

$$\mathbf{R}_{\hat{i}_{01}^{\perp},\phi}\mathbf{R}_{\hat{i}_{01},\theta}\mathbf{R}_{\hat{z},\psi}. \tag{2.81}$$

As with Ullman's method (Section 2.9.2), we do not know the signs of $\sin\theta$ and $\sin\phi$, but only that inverting both signs simultaneously corresponds to the reflective ambiguity.

2.9.5 Summary of the three computations

Here I summarize how each method can be used to compute 3D pose from three corresponding points. To begin, transform the model and image points so that (1) the model points lie in the image plane, (2) $\vec{m}_0$ and $\vec{i}_0$ are at the origin of the image coordinate system, and (3) $\vec{m}_1 - \vec{m}_0$ and $\vec{i}_1 - \vec{i}_0$ lie along the x axis. Then use one of the three methods to compute the scale factor and out-of-plane rotation, as follows:

- Ullman's method
  1. Use Equations 2.52-2.54 to get a, b, and c.
  2. Substitute a, b, and c into Equation 2.55 to get s.
  3. Calculate $\cos\phi = \frac{x_1}{s\bar{x}_1}$ and $\cos\theta = \frac{y_2}{s\bar{y}_2}$.
  4. Calculate $\sin\phi = \sqrt{1-\cos^2\phi}$ and $\sin\theta = \sqrt{1-\cos^2\theta}$.
  5. Construct the rotation matrix $\mathbf{R}$ using Equation 2.47.
- Huttenlocher and Ullman's method
  1. Solve Equations 2.56 and 2.57 for l_{11} and l_{12}, and Equations 2.58 and 2.59 for l_{21} and l_{22}.
  2. Let $\vec{e_1} = (0,1)$ and $\vec{e_2} = (1,0)$. (Any orthogonal, equal-length vectors can be used.)
  3. Use Equation 2.60 to get $\vec{e_1}'$ and $\vec{e_2}'$.
  4. Substitute $\vec{e_1}'$ and $\vec{e_2}'$ into Equations 2.61 and 2.62 to get k_1 and k_2.
  5. Substitute k_1 and k_2 into Equation 2.64 to get c_1.
  6. If $c_1 \neq 0$, calculate $c_2 = \frac{k_1}{c_1}$. Otherwise get c_2 from Equation 2.65.
  7. Use Equation 2.67 to get s.
  8. Use Equation 2.66 to get $s\mathbf{R}$. Divide through by s if $\mathbf{R}$ is desired instead of $s\mathbf{R}$.
- Grimson, Huttenlocher, and Alter's method
  1. From the model points, compute R_{01}, R_{02} and ξ, and, from the image points, compute d_{01}, d_{02}, and ω.
  2. Use Equation 2.76 to get t.
  3. Use Equation 2.77 to get $\cos\phi$.
  4. Use Equation 2.70 to get s.
  5. Calculate $\sin\phi = \sqrt{1-\cos^2\phi}$.
  6. Use Equations 2.79 and 2.80 to get $\cos\theta$ and $\sin\theta$.
  7. To transform any point $\vec{p}$, substitute $\cos\phi$, $\sin\phi$, $\cos\theta$, $\sin\theta$, and $\vec{p}$ into Rodrigues' formula, Equation 2.68, to get $\mathbf{R}\vec{p} = \mathbf{R}_{\widehat{i_{01}^{\perp}},\phi}\mathbf{R}_{\widehat{i_{01}},\theta}\vec{p}$.

2.10 Conclusion

The weak-perspective three-point problem is fundamental to many approaches to model-based recognition. In this chapter, I illustrated the underlying geometry, and then used it to derive a new solution to the problem, and to explain the various special cases that can arise; the special cases determine when there are zero, one, and two solutions. Also, I discussed earlier solutions to the problem in detail.

The new solution is based on the distances between the matched model and image points, and is used to obtain an expression for a direct alignment of a model to an image. As a result, the solution given here may be easier to use, and, for recognition systems that repeat the computation of the model pose many times, may be more efficient.

Furthermore, the geometric approach in this chapter provides additional insights into the problem. First, it was demonstrated that the pose solution may be unstable either when the model points are nearly collinear or when the plane of the matched model points is parallel to the image plane. This property is not particular to the pose solution given here, but is inherent in the underlying geometry. Second, this chapter resolves which solution to Ullman's original biquadratic is correct, and, specifically, showed that the false solution corresponds geometrically to inverting the roles of the model and image points. Also, this chapter makes evident the symmetry of the solution with respect to the ordering of the points. In general, the geometric approach has been useful in gaining understanding of the basic problem, and may prove useful for providing insights when a related problem needs to be solved.

Chapter 3

Uncertainty in Point Features

As discussed in Chapter 1, features derived from an image invariably contain errors. The approach in Section 1.3 uses triples of matched point features to generate hypotheses, and then uses model features such as points, line segments, and curve segments for deciding which hypotheses are correct. To decide correctness, the algorithm uses the matched point triples to predict the image locations of the model features (step 2b). Errors in the locations of the image points, however, lead to uncertainty in the predicted locations of these model features. Consequently, in step 2c of the algorithm, the hypothesized three-point match is used to compute search regions for finding matches to the model features.

In the past, to account for the uncertainty, people tried considering all image features as candidate matches [Grimson84]; however, the combinatorics of such an approach are prohibitive [Grimson90a]. In addition, people tried looking for matches in a region of fixed size and shape about each predicted feature [Huttenlocher88], but this assumes the size and shape do not significantly change. If this assumption is wrong, it can lead to correct hypotheses being discarded and incorrect hypotheses being accepted; occurrences of which are known as *false negatives* and *false positives*, respectively.

Using a standard model for error in the image points (Section 3.1), this chapter shows that the shapes of the uncertainty regions for point features generally do not change, but the sizes can change considerably. Further, it is demonstrated that the uncertainty regions generally are circular (Section 3.2), and a method is given for fitting "uncertainty circles" to them (Section 3.4). In addition, the situations where the uncertainty regions are not circular are described (Section 3.3). Lastly, the uncertainty circles are compared to previous uncertainty propagation techniques (Section 3.5).

3.1 Bounded Error Model

The errors in the image points arise from several sources, including the optics used to obtain the image, the edge detector, and the process for finding distinguishing points from edges. The effect of these errors is to alter the locations of the image points. These locations will be altered by at most some amount ϵ, which empirically seems to be about five pixels. Circles of radius ϵ can then be used to bound the error in the sensed or *nominal* locations of the image points. This approach to modelling error is known as a "bounded error model," and has been used often for performing robust recognition [Grimson84] [Baird85] [Ellis87] [Cass90] [Jacobs91].

3.2 Uncertainty Circles for Bounding Uncertainty Regions

To see how well uncertainty circles do for bounding the errors in the image locations of predicted model points, this section runs two experiments that compare the true regions to the circular fits. The radii of the circles is computed by taking the maximum distance from the nominal point to a point on the boundary. To compare the regions, we need a measure of error between the true region and the approximation. When the circular approximations are poor, the circles will badly over-bound the true regions. One measure is what fraction of the circle the true region leaves uncovered. Let A_t equal the area of the true region and let A_c equal the area of the approximating circle. Then the fraction just mentioned is given by $\frac{A_c - A_t}{A_c}$. When the approximation is good, however, we want to know the relative error from the true value, which is given by $\frac{|A_t - A_c|}{A_t}$. Using the fraction of the area when the uncertainty circle over-bounds the true region and using the relative error when it does not, the error measure is

$$\begin{cases} \frac{A_c - A_t}{A_c} & \text{if } A_c \geq A_t, \\ -\frac{|A_t - A_c|}{A_t} & \text{otherwise,} \end{cases}$$

where the sign is used to discriminate the two cases. This expression is the same as

$$\frac{A_c - A_t}{\max(A_c, A_t)} \,. \tag{3.1}$$

In this expression, the max is needed even though in theory $A_c > A_t$, because the method used below to compute A_t can overestimate it a little when the circular approximation is good.

Experiment 1: Accuracy of uncertainty circles for random models

If we are using uncertainty circles in a recognition system, we wish to know how often to expect the uncertainty circles to be correct. In terms of the error measure (Equation 3.1), we wish to know what percent of the time the maximum value of the error measure will be at most, say 1%, or 10%. Conversely, what will be the maximum error 90% of the time, 95% of the time, or 99% of the time?

To estimate these numbers, I ran a series of trials to simulate an actual system and computed the error measure for each. The percent of time the error measure is expected to satisfy some criteria is estimated by the fraction of trials over which it satisfies that criteria. For an actual system, I consider an alignment algorithm that selects triples of points from an image and matches them to triples of points from a model. I assume that the points in the image effectively arise at random, which is reasonable if the image contains significant clutter.

Method

This experiment runs one hundred trials where a model is projected into an image and the error measure of Equation 3.1 is computed for each model point. In each trial, a random triple of image points is matched to a random triple of model points taken from a randomly-generated model (see Appendix C for details). The three-point match is used to project the model into the image, which gives the nominal image locations of the model points. As described in Chapter 2, except for model points in the plane of the matched model points, there are two possibilities for each nominal image location.

Using $\epsilon = 5$, the ϵ-circles around the three image points are sampled uniformly at twenty-five points each (see Fig. 3-3). Every triple of points between the samples on the uncertainty circles is matched to the three model points. Each match is used to compute the image locations of all the model points. This results in a set of boundary and interior points for uncertainty regions. The area of each region is computed by scanning its points into an image and counting the number of pixels within the resulting image boundary (this step is explained further in Appendix D). There are two sets of boundary points corresponding to the two weak-perspective solutions (Chapter 2), which results in two areas per uncertainty region (see Appendix D).

Given the boundary points for an uncertainty region, the radius of the corresponding uncertainty circle is obtained by taking the maximum distance from the nominal point to a boundary point. For a radius r, the area of the circle is πr^2.

Results and Discussion

Over the 100 trials, 1163 uncertainty regions were tested. The average area was 583.53 for the correct uncertainty regions and 662.43 for the approximating circles. Fig. 3-1 shows a histogram of the percent errors in the circular approximation (using the error measure). The largest peak of the histogram is at 0. The average percent error is -10.82, the median is between -11 and -12, and the range is $[-35.11, 81.65]$. Negative errors occur because, when the fit is good, the method used to compute the true regions may actually overestimate them a little (Appendix D). The large negative errors are all for situations where the circles are very small (radii between five and eight pixels); the error measure is sensitive to these cases because of the slight overestimation in the method for computing regions. The errors of particular concern are large positive errors, which arise when the uncertainty circles are large overestimates. As will be seen next, such errors occur rarely.

By summing up to an index in the histogram and then dividing by the total number of entries, we get the fraction of time that the error was less than that index. Doing so gives that 96.73% of the time the error between the true region and the approximation was less than 1%, and 97.94% of the time the error was less than 10%. Conversely, the maximum error 90% of the time was 1%, 95% of the time it was also 1%, 98% of the time it was 10%, and 99% of the time it was 51%. These results suggest that uncertainty circles are generally very accurate.

Experiment 2: Accuracy of the uncertainty circles for the telephone model

The experiments on random models indicate that for most objects the circular approximations are good. To see how accurately random models reflect true objects, I took a model of typical object for the system to handle, a telephone (Fig. 3-2), and re-ran the same set of trials. The telephone model was built by hand. The model points are listed in Table 3.1. The first eight points were measured in inches on the actual object, and the last two were added to make the appearance of the model more complete.

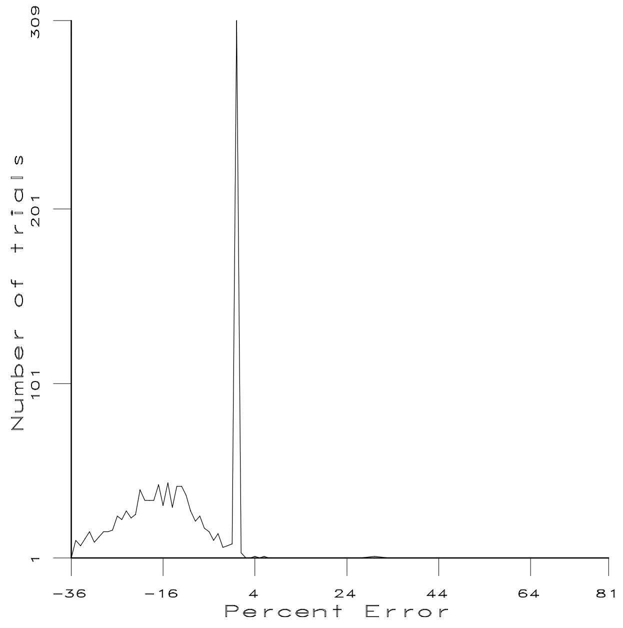

Figure 3-1: Histogram for 1163 uncertainty regions using random models

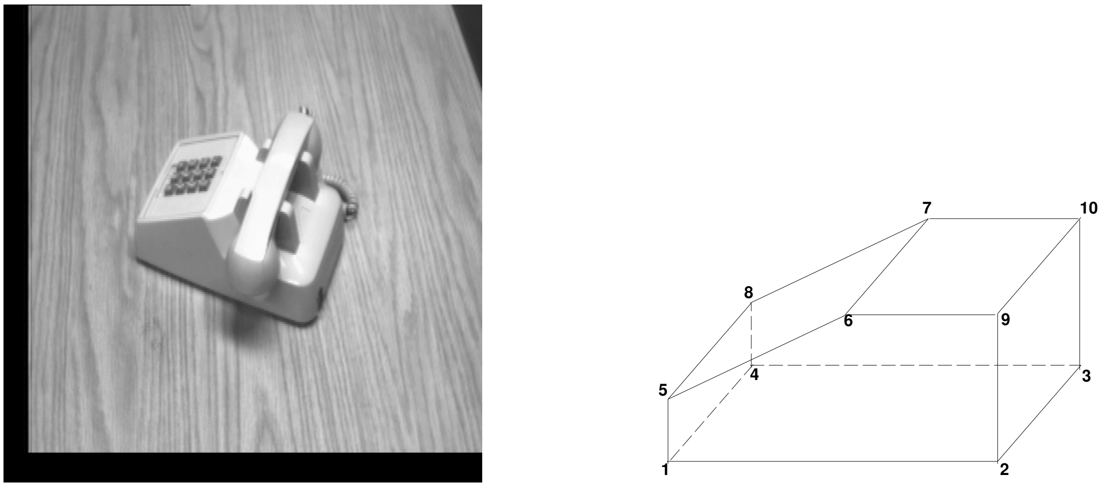

Figure 3-2: A telephone and a model of a telephone

	x	y	z
1	0	0	0
2	9	0	0
3	9	4.625	0
4	0	4.625	0
5	0	0	1.625
6	3.5	0	3.5
7	3.5	4.625	3.5
8	0	4.625	1.625
9	9	0	3.5
10	9	4.625	3.5

Table 3.1:

Points on the telephone model.

Method

The method is the same as in Experiment 1, except that the telephone model was used at every trial instead of a new, random model (Fig. 3-3).

Results and Discussion

For 100 trials with the phone model, 1092 uncertainty regions were generated. The average area was 495.59 for the correct uncertainty regions and 450.13 for the approximating circles. Notice that this time the average area for the overestimates is lower than for the exact areas. This is because, as mentioned earlier, the method used to compute the true regions can overestimate them a little when the fit is good (Appendix D). This effect turned out to be stronger than the overestimate in the circular fit, because very few of the circular fits were poor.

The resulting histogram for the phone model is shown in Fig. 3-4, overlayed with the histogram for random models. The distributions are similar, with the phone model having a smaller range of values. The average percent error is -10.72, the median is between -11 and -12, and the range is $[-31.37, 27.02]$. Observe that the average and median errors are very close to those for random models.

For the phone model, 98.01% of the time the error between the true region and the approximation was less than 1%, and 99.08% of the time it was less than 10%. As before, the maximum error 90% and 95% of the time was 1%. This time, however, the maximum error 98% was also 1%. Further, 99% of the time the maximum error was 10% instead

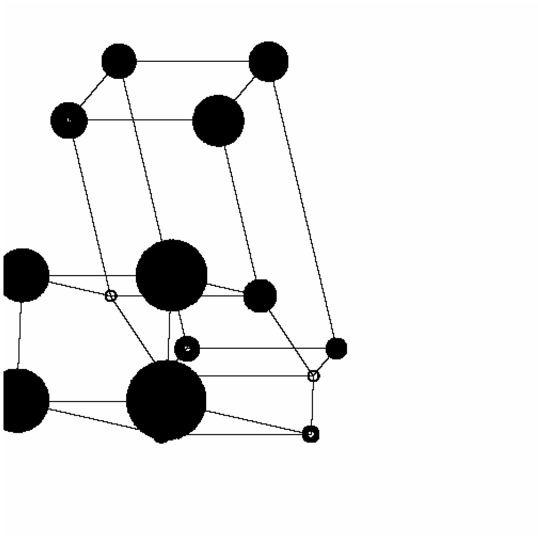

Figure 3-3: Propagated uncertainty regions for a telephone model (both solutions are shown). The two small, unfilled circles are sampled ϵ-circles (the third one is occluded). Observe that the propagated regions appear circular.

of 51%. So it appears the circular fits work better for the specific model of a telephone.

3.3 Cases Where Errors Are Greatest

This section looks closely at the cases where the errors are large. Doing so may help to infer the situations where circles are poor approximations, which is important for knowing when the uncertainty circles badly overestimate the true regions. Also, knowing when the approximation breaks would allow for avoiding these cases or handling them specially.

Of the one hundred trials on random models, there were two which had errors greater than 25%. For each trial, Fig. 3-5 displays the uncertainty region and uncertainty circle that had the largest error. For one trial, the largest error was 78.8%. The model point with this error had extended affine coordinates $(2.037, -2.227, .01368)$ (extended affine coordinates were defined in Chapter 2). For this trial, the three matched model points were $(15, -74, -112)$, $(-48, 57, -7)$, and $(-3, 59, -70)$, and the three matched image points were $(296, 416)$, $(132, 230)$, and $(120, 336)$. More interestingly, the angles between

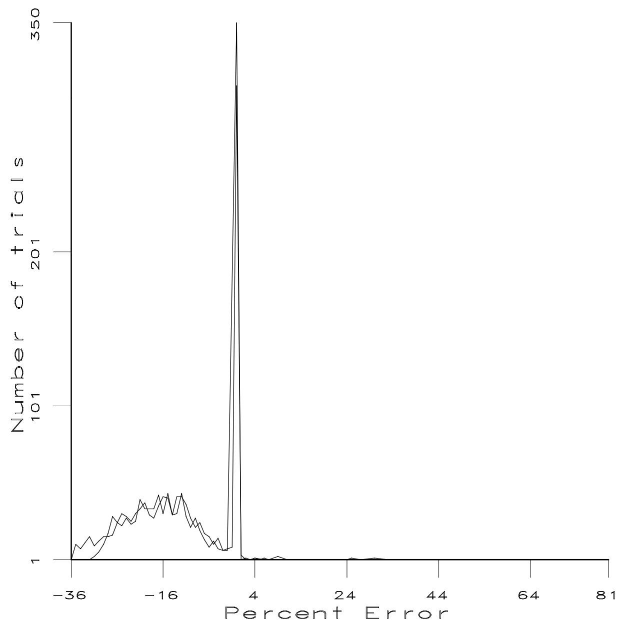

Figure 3-4: Histograms for 1163 uncertainty regions using random models and 1092 uncertainty regions using telephone model.

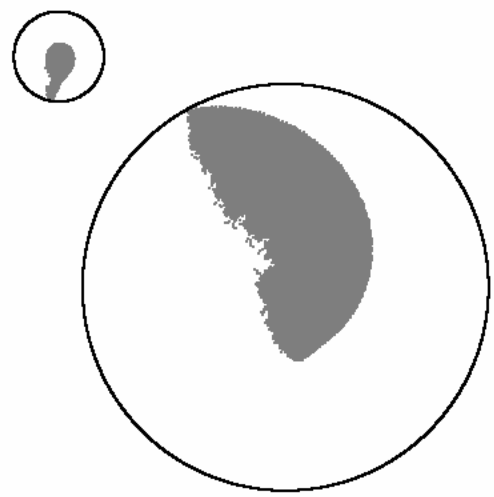

Figure 3-5: Largest errors for random models: 81.7% (upper left), 78.8% (bottom right)

these points are

$24.39°$, $107.03°$, $48.58°$	for the model points
$24.15°$, $107.99°$, $47.86°$	for the image points

Notice that these angles are very close. Geometrically, this means the plane of the model points is almost parallel to the image, a situation which Chapter 2 warned was unstable.

For the other trial, the largest error was 81.7%, and the extended affine coordinates of the corresponding model point were $(-.7151, 1.404, .002413)$. The three matched model points were $(9, -19, 170)$, $(-3, 35, 6)$, and $(-83, 2, 57)$. The three matched image points were $(272, 191)$, $(34, 198)$, and $(101, 314)$. The angles between the points are

$35.40°$, $86.50°$, $58.10°$	for the model points
$34.04°$, $84.28°$, $61.67°$	for the image points

Again the angles are very close, which means the plane of the matched model points is almost parallel to the image. These cases suggest that we should be cautious with the

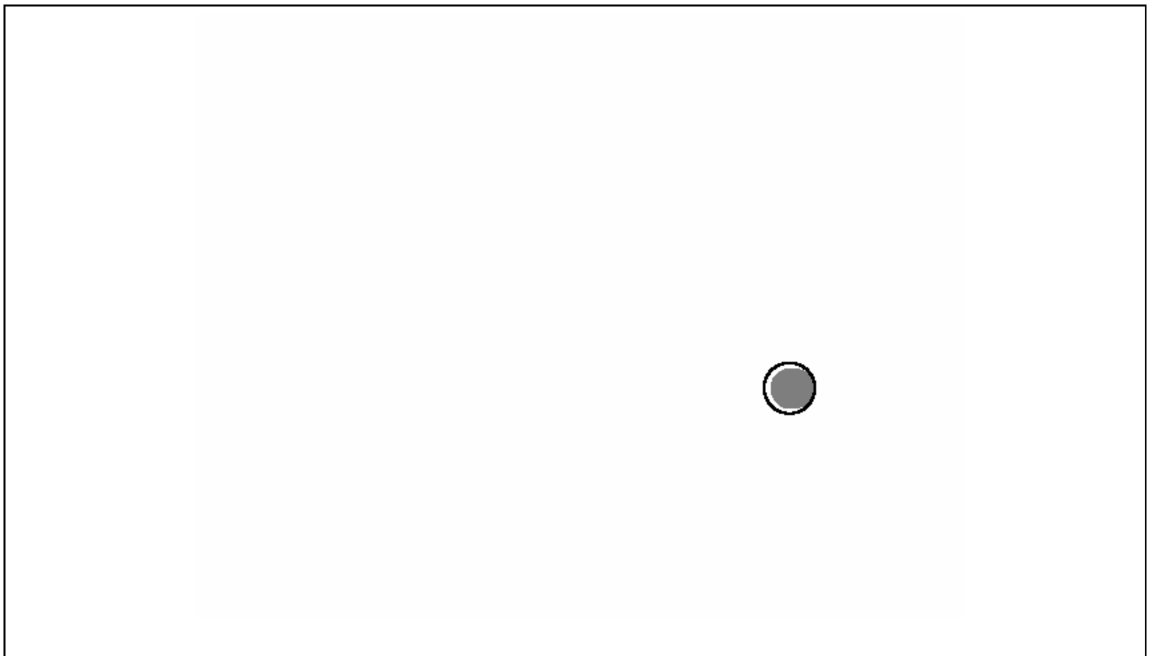

Figure 3-6: Largest error for the telephone model: 27.0%

uncertainty circles when the model plane is nearly parallel to the image.

The true uncertainty regions pictured in Fig. 3-5 have strange shapes. The concavity in the larger region is due to the interior of the region not being filled, which is a result of sampling only the boundaries of the error regions of the matched image points. Ignoring the concavity, there is an almost straight line bounding part of the region. The source of this line is the way the uncertainty regions are computed. As explained in Appendix D, the propagated points are separated into two groups in order to handle the two solutions for pose (see Chapter 2). The points are separated according to whether H_1 or H_2 from the pose solution is positive or negative. For Fig. 3-5, if all the points from both solutions were plotted, then a smoothly curved boundary for the entire region could be expected.

For the phone model, there was only one trial out of the one hundred which had errors greater than 25%, and the largest error in this case was 27.0%. The uncertainty region and uncertainty circle are shown in Fig. 3-6. The extended affine coordinates of the point with largest error were $(.02475, .3641, -.000032)$, and the angles between the model and image points were

Figure 3-7: Smallest Errors: -31.4% and -35.1%

$32.80°$, $57.20°$, $90.00°$	for the model points
$77.07°$, $61.36°$, $81.57°$	for the image points

Two of the angles are close, but not as close as they were for random models. At the same time, the worst-case error is not nearly as bad as for random models.

Fig. 3-7 displays the regions with the largest negative errors for the trials on random models and the phone model. Recall that negative errors arise because there may be extra pixels counted along the boundaries of the true regions when computing the areas (see Appendix D). From the figure, negative errors can be as small as -35% and the approximation visibly be good.

In summary, we can infer that, in an alignment system that tries many or all pairs of point triples for aligning a model to the image, situations with large errors could be avoided by checking whether the angles between the points are similar. However, this may lead to relying on an arbitrary threshold. Consequently, it perhaps would be better to handle these cases specially by using another technique such as that used in the experiments, namely, to sample extensively and then walk the boundaries of the resulting regions.

3.4 Computing Uncertainty Circles

Given that circles centered at the nominal points approximate well the uncertainty region boundaries, all that is needed is to compute the radii of the circles. Since only one boundary point is needed to compute the radius, a straightforward approach is to sample points from the error circles around the matched image points and take the maximum distance from the nominal point as the radius. This process will be efficient if few sample points are required.

Experiment 3: Using fewer sample points for random models

To see how few sample points are needed, this experiment tests, for various numbers of points, n, and for a series of trials, the percent of time (fraction of trials) that the error in using n points instead of twenty-five is less than some limit. Twenty-five is the number of points used in the last two experiments.

Method

A series of one hundred trials are run using random image triples matched to random model triples from randomly-generated models, using the same method as in Experiment 1. For each trial, the error circles around the matched image points are sampled uniformly at twenty-five points and ten points. For each propagated uncertainty region, the error in using the smaller number of samples to using twenty-five samples is computed. This is repeated for nine, eight, and seven sample points as well.

Results and Discussion

The results are shown in Table 3.2. It may be observed that the percentages do not strictly decrease as fewer sample points are used. This can be explained by the fact that the circles around the image points are sampled uniformly, so that using different numbers of sampled points can give different samples on the circles. Consequently, when the percentages are close, there may be cases where fewer sample points do better. Nevertheless, this effect should be small. Notice that the average percent error does indeed increase monotonically.

We can use Table 3.2 to pick a reasonable number of points for sampling the image error circles. From the table, if we permit 5% error, then using eight sample points instead of twenty-five can be expected to be accurate over 99% of the time. Also, the average error in using eight points is very small (1.137%).

A better feel for how accurate is the use of fewer sample points is given by statistics

	1%	2%	3%	4%	5%	6%	ave	max	min
10	72.06	93.12	98.61	99.13	99.22	99.56	0.684	21.97	-.34
9	57.27	84.16	98.00	98.70	98.87	99.04	1.031	31.60	-.37
8	55.61	75.98	90.43	98.87	99.39	99.57	1.137	17.12	-.49
7	46.30	63.27	77.89	91.65	97.91	98.61	1.670	25.53	-.31

Table 3.2:

Percentage of time error was less than 1%-6% for different numbers of sample points. Also shown are the average, maximum, and minimum percent errors over all the trials. Results are based on 1149 propagated uncertainty regions using random models.

	ave	max	min	ave percent	max percent	min percent
10	.05	2.55	-.05	.344	11.67	-.17
9	.08	3.87	-.03	.521	17.30	-.18
8	.08	3.24	-.05	.573	8.96	-.24
7	.13	4.21	-.02	.844	13.70	-.16

Table 3.3:

Differences in radii for different numbers of sample points. Results are based on 1149 propagated uncertainty regions using random models.

on the radii, shown in Table 3.3. From the table, the average difference in the radii for eight sample points was .08 pixels, and the worst case difference was 3.24 pixels. Relative to the radius for twenty-five points, the average difference is .573%, and the maximum difference is 8.96%.

Experiment 4: Using fewer sample points for telephone model

Method

This experiment is the same as Experiment 3, except that the phone model is used instead of random models.

Results and Discussion

Tables 3.4 and 3.5 give the results. From Table 3.4, we again can use eight points to limit errors to 5% over 99% of the time. From observing both tables, it appears that using fewer sample points works slightly better with the phone model than with random models.

To illustrate the use of uncertainty circles, Fig. 3-8 shows an example of the propa-

	1%	2%	3%	4%	5%	6%	ave	max	min
10	66.40	91.91	99.37	99.55	99.64	99.64	0.726	8.55	-.33
9	60.83	84.19	98.02	99.46	99.55	99.55	0.913	13.19	-.33
8	62.15	91.91	99.37	99.55	99.64	99.64	0.981	11.76	-.30
7	46.46	64.24	80.32	92.81	98.38	99.55	1.532	12.80	-.33

Table 3.4:

Percentage of time error was less than 1%-6% for different numbers of sample points. Also shown are the average, maximum, and minimum percent errors over all the trials. Results are based on 1113 uncertainty regions using the telephone model.

	ave	max	min	ave percent	max percent	min percent
10	.05	0.69	-.05	.365	4.37	-.17
9	.06	1.30	-.02	.459	6.83	-.17
8	.07	1.12	-.03	.494	6.07	-.25
7	.10	1.23	-.03	.772	6.62	-.16

Table 3.5:

Differences in radii for different numbers of sample points. Results are based on 1113 uncertainty regions using the telephone model.

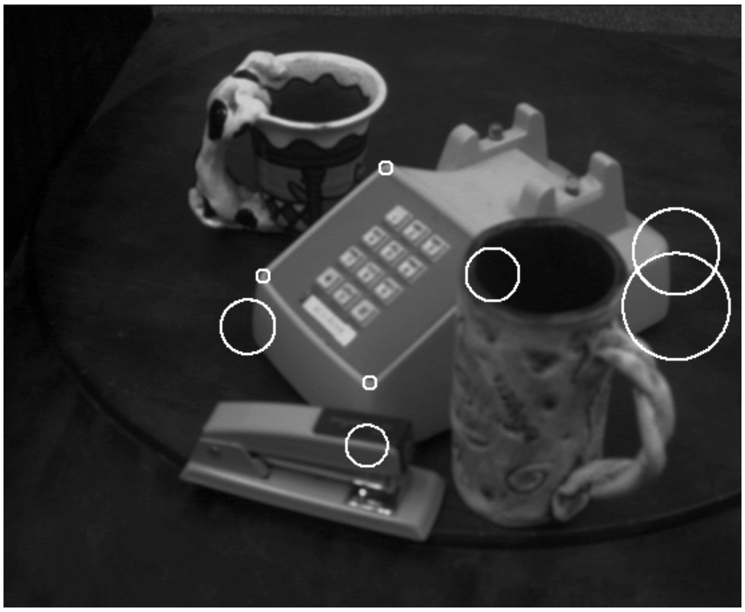

Figure 3-8: Propagated uncertainty in a real image, which was provided by David Jacobs. The three smallest circles correspond to assumed errors in the matched image points, and, given those errors, the larger circles show the sets of possible locations of the other corner points of the telephone.

gated uncertainty circles, where eight sample points were used. The three smallest circles correspond to the assumed errors in the matched image points, which in this example were matched correctly. For the unmatched model points, the other circles show the regions to be searched for matching image points. The self-occluded model points were removed beforehand. Still, some of the remaining corner points are occluded by other objects, and the uncertainty regions provide a means to reason that this is so after a relatively small amount of search in the image.

Notice that the sizes of the propagated uncertainty regions vary considerably for different model points. Consequently, an approach that relies on fixed-sized error bounds, as in [Huttenlocher88], can lead to correct matches being missed (when the bounds are too small), and incorrect matches being accepted (when the bounds are too large and include spurious image points).

Method	Models	Type	$\overline{\mu}$
Uncertainty Circles	Random	Solid	.003722
Uncertainty Circles	Phone	Solid	.003747
Bounding Polygons	Random	Solid	.008279
Exact Circles	Random	Planar	.002783

Table 3.6:

Expected selectivities of point features.

3.5 Expected Selectivity of Point Features

The probability that a feature distributed randomly over an image falls into an uncertainty region is known as the *selectivity* of the region [Grimson92b]. This quantity is useful for analyzing the reliability of recognition systems [Grimson92a] [Grimson92b], including, as will be seen in Chapters 5 and 6, the system proposed here. For point features, the selectivity is the area of the region divided by the image area, $\frac{A}{A_I}$, where the area of the region takes into account the uncertainty in the unmatched image points by expanding the propagated region outwards by ϵ.

In the past, the concept of selectivity has been applied to alignment where the models are flat [Grimson92b], and also to alignment with solid models but using a different uncertainty propagation technique [Grimson92a]. When the models are flat, the propagated uncertainty regions can be computed exactly. It would be interesting to see how much the chance of a false positive increases from planar to solid models. Also, it would be useful to know how the uncertainty propagation technique used here compares to the one in [Grimson92a]. We can use the expected selectivity to make these comparisons.

Experiments 5 and 6: Expected selectivity of point features

Method

To compute the expected selectivity, I re-ran 1000 trials of the same type as in Experiments 3 and 4, except five was added to each radius before computing the area, in order to account for expanding the uncertainty region outwards by $\epsilon = 5$ pixels.

Results and Discussion

Using random models with eight sample points over 1000 trials gave 11349 propagated regions with average area 973.25 square pixels. Using the phone model with eight sample points over 100 trials gave 11085 propagated regions with average area 979.78 square pixels. For an image of size 454×576, the resulting selectivities along with those for [Grimson92a] and [Grimson92b] are shown in Table 3.6. The expected selectivity for the uncertainty circles is about half that for [Grimson92a], which implies that the uncertainty circles should give significantly better performance. Furthermore, it appears that the selectivities of solid models are only slightly greater than for planar ones. We can infer from this that, when point features are used, recognizing solid objects with alignment is a only a little more sensitive to false positives than recognizing planar objects.

Chapter 4

Uncertainty in Line Features

The preceding chapter dealt with uncertainty in the predicted locations of point features (step 2c of the alignment algorithm of Section 1.3). A set of distinguished points, however, usually is not reliable at identifying a model in an image. Consequently, recognition systems often use more extended features such as line segments for verification [Bolles82] [Goad83] [Lowe85] [Ayache86] [Horaud87] [Huttenlocher90]. This chapter extends the uncertainty analysis of the preceding chapter to line features (Section 4.1). Furthermore, a formula is derived for selectivity for line features (Section 4.2). The selectivity for lines is demonstrated to be significantly less than for points (Section 4.3).

4.1 Line Uncertainty Regions

Section 3.2 showed how to compute uncertainty circles to bound the propagated uncertainty in predicted model points. We can use this result to bound the uncertainty in predicted model line segments. First, for each model line segment, calculate the uncertainty circles for its endpoints. Next, if we ignore fragmentation and partial occlusion, an overestimate of the set of image line segments that could match a model line segment is given by the set of all line segments connecting pairs of points in the two circles. To then allow for some fragmentation and occlusion, we would also accept any sub-segment of one of these line segments. We can find all candidates for a given model segment by first gathering all image line segments that lie entirely within the uncertainty region formed by the uncertainty circles and their common outer tangents (see Fig. 4-1). Then we will

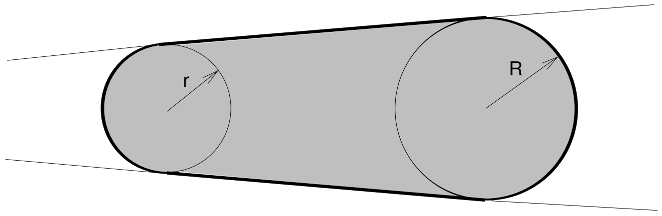

Figure 4-1: Region to search for candidate line segments

keep only the line segments that can be extended to intersect both of the uncertainty circles.

4.2 Selectivity of Line Features

The selectivity of a line uncertainty region is the chance that a spurious line segment randomly falls into it. Ideally, the line selectivity could be estimated by the chance that the endpoints of a random line segment fall within the point uncertainty regions of a predicted model segment's endpoints. With fragmentation and occlusion, however, the endpoints of the corresponding image segment may not appear in those regions. To allow for either endpoint to be occluded, the last chapter treated every model point independently. By so doing, at least one of the endpoints is required to be unoccluded. In addition, the constraint from the orientation of a model segment is lost. Instead of looking for endpoints, we can look for pieces of the predicted model segments, as described in Section 4.1. If pieces of line segments are used, which still constrain the orientation and partially constrain the length, the selectivity for line segments can be expected to be much less than for points.

4.2.1 Non-overlapping uncertainty circles

This section considers the case in which the uncertainty circles for the endpoints do not overlap, which is the most common situation. Consider an image segment of known length and orientation. There is a set of translations that place the segment within the image. The line selectivity equals the fraction of these translations that place the segment within the line uncertainty region. As a note, the set of translations of a line

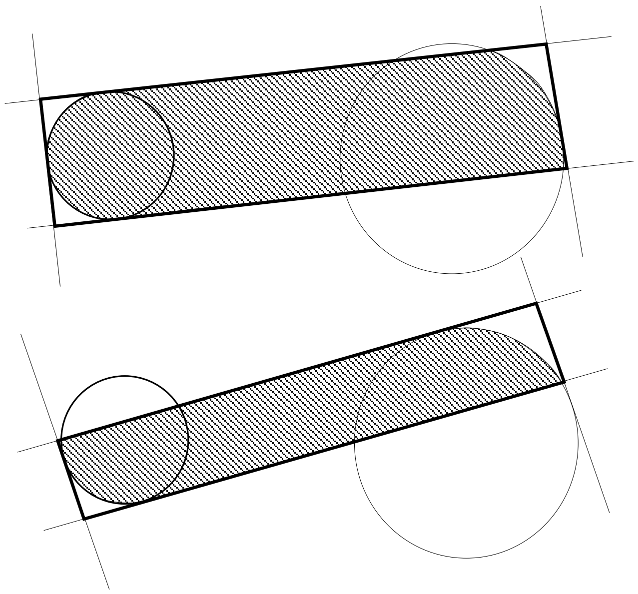

Figure 4-2: Region of translations with orientation constraint and rectangular upper bound.

segment of known length and orientation is the same as its *configuration space* [Lozano-Pérez87], since a translation determines the position of every point on the line segment. The configuration space of an image segment with respect to an uncertainty region can be obtained by shrinking the region along the segment's orientation and by its length.

Examples of the constraint from an image segment's orientation are illustrated by the shaded regions in Fig. 4-2. The figure shows two cases, distinguished by the orientation of the image segment relative to the orientation of the common outer tangent, which from Fig. 4-3 is given by

$$\theta_1 = \sin^{-1} \frac{R - r}{L} \tag{4.1}$$

As shown in Fig. 4-4, the orientation of an image segment within the uncertainty region is bounded by the orientations of the common crossed tangents of the uncertainty circles. Letting θ_2 be the maximum allowed orientation of a candidate image segment, from the figure

$$\theta_2 = \sin^{-1} \frac{R + r}{L} \tag{4.2}$$

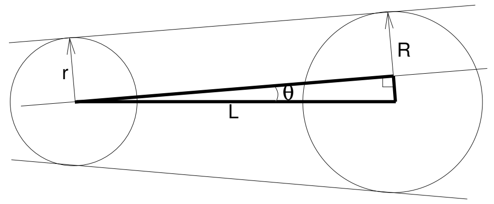

Figure 4-3: Orientation of the common outer tangents to the circles.

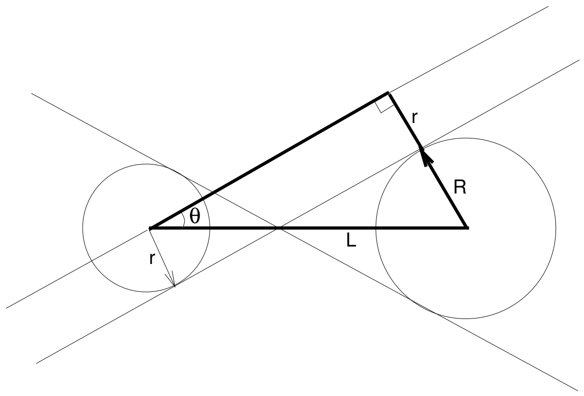

Figure 4-4: Orientation of the common crossed tangents of the circles. This is the maximum possible angle of a line segment with an endpoint in each circle.

Note that θ_1 exists iff $L \geq R - r$, and θ_2 exists iff $L \geq R + r$. If the uncertainty circles do not overlap, then $L \geq R + r$.

Starting from the region of translations with orientation constraint, a set of translations with length constraint also is obtained by shrinking the shaded region in Fig. 4-2 by the length of the image segment. The area of the region can be computed by moving the image segment perpendicular to its orientation, as shown in Fig. 4-5, parameterized by u. The area is given by summing the distances between (x_1, y_1) and (x_2, y_2) over the range of u. Let ℓ be the length of the image segment. Appendix E.1 shows the area is given by,

$$A = \begin{cases} \int_{-r}^{u_{\max}} \left(-\ell + L\cos\theta + \sqrt{R^2 - (u + L\sin\theta)^2} - \sqrt{r^2 - u^2}\right) du & \\ & \text{if } L\sin\theta \leq R + r, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

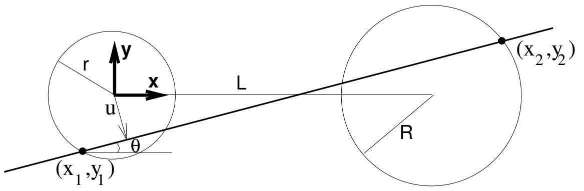

Figure 4-5: Computing the area of a region of translations for an image line segment of known length and orientation.

where $u_{\max}$ involves a solution to a quadratic equation whose coefficients are given by complicated expressions.

Computing the line segment selectivity with this formula is messy, and so instead I compute a close overestimate. Fig. 4-6 shows a rectangular box which can be used to bound the range of translations. For comparison, Fig. 4-2 shows the rectangular box surrounding each corresponding line uncertainty region. From Fig. 4-6, the base of the rectangle is $R + r + L\cos\theta$. Further, the height of the rectangle is $2r$ for the top picture where $0 \leq \theta \leq \theta_1$, and $R + r - L\sin\theta$ for the bottom picture where $\theta_1 \leq \theta \leq \theta_2$. Observe that for an image segment of length ℓ to fit in the rectangle, ℓ must be less than or equal to the base, $R + r + L\cos\theta$. After shrinking the rectangle along the base by ℓ, the area of the region is

$$A = \begin{cases} (R + r + L\cos\theta - \ell)2r & \text{if } \theta \in [0, \theta_1], \quad \ell \leq R + r + L\cos\theta, \\ (R + r + L\cos\theta - \ell)(R + r - L\sin\theta) & \text{if } \theta \in [\theta_1, \theta_2], \quad \ell \leq R + r + L\cos\theta, \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

Note that $R + r - L\sin\theta \geq 0$, since $\theta \leq \theta_2 = \sin^{-1}\frac{R+r}{L}$.

With respect to the image, Fig. 4-7 shows that the area of translations for the same image segment is

$$A_I = (w - \ell\cos\theta_I)(h - \ell\sin\theta_I) \tag{4.5}$$

The selectivity of a random line segment of known length and orientation is $\frac{A}{A_I}$.

In general, there will be several line segments that fall within a line uncertainty region, and the line segments will have different lengths and orientations. To account for orientation, we can assume that random line segments are equally likely to fall at any angle. Then we can integrate the formulas for A and A_I over their respective ranges

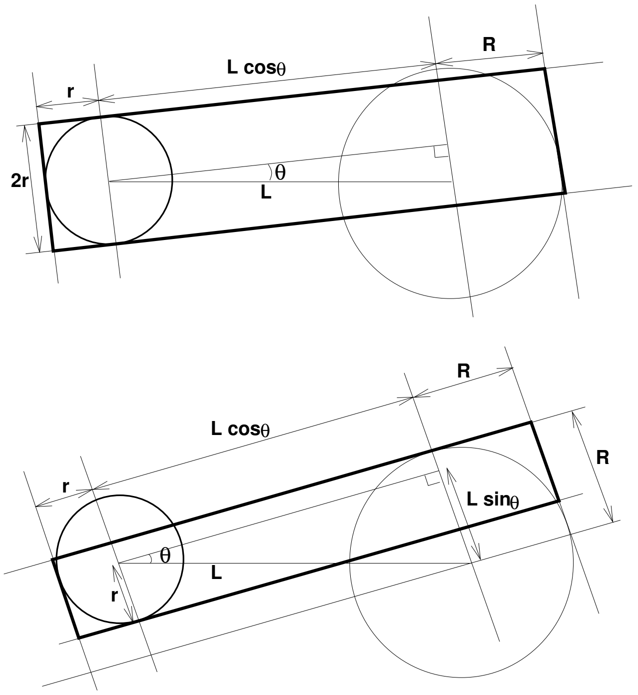

Figure 4-6: Dimensions of the rectangular upper bound on the region of translations of a line segment with orientation constraint.

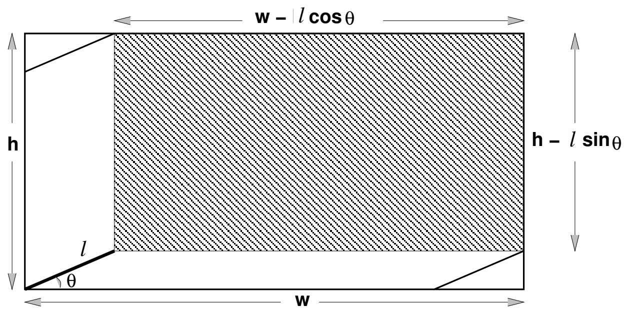

Figure 4-7: Region of translations of a line segment over an image.

of allowable orientations to get volumes of allowable positions of a random line segment (with known length). Integrating the two area expressions in Equation 4.4 over an arbitrary range $[\omega_1, \omega_2]$ gives two corresponding volume expressions (see Appendix E.2):

$$\begin{aligned} v_1(\omega_1, \omega_2) &= (R + r - \ell)2r(\omega_2 - \omega_1) + 2rL(\sin\omega_2 - \sin\omega_1) & (4.6) \\ v_2(\omega_1, \omega_2) &= (R + r - \ell)(R + r)(\omega_2 - \omega_1) + (R + r - \ell)L(\cos\omega_2 - \cos\omega_1) \\ &\quad + (R + r)L(\sin\omega_2 - \sin\omega_1) - \frac{1}{2}L^2(\sin^2\omega_2 - \sin^2\omega_1) & (4.7) \end{aligned}$$

From Equation 4.4, the range of θ is divided into two intervals at $\theta = \theta_1$. Also in Equation 4.4, the length of the image segment constrains the range of orientations such that $\ell \leq R + r + L\cos\theta$, or equivalently, $\cos\theta \geq \frac{\ell - (R+r)}{L}$, or $\theta \leq \phi$, where

$$\phi = \cos^{-1}\left(\frac{\ell - (R + r)}{L}\right). \tag{4.8}$$

Note that ϕ exists iff $R + r - L \leq \ell \leq R + r + L$. The first inequality holds since the circles do not overlap, and the second must be true for the image segment to fit in the uncertainty region (Fig. 4-1). From these constraints, the volume V that corresponds to the area A in Equation 4.4 is given by

$$V = 2\begin{cases} v_1(0, \phi) & \text{if } \phi \leq \theta_1, & l \leq R + r + L, \\ v_1(0, \theta_1) + v_2(\theta_1, \phi) & \text{if } \theta_1 \leq \phi \leq \theta_2, & l \leq R + r + L, \\ v_1(0, \theta_1) + v_2(\theta_1, \theta_2) & \text{if } \theta_2 \leq \phi, & l \leq R + r + L, \\ 0 & \text{otherwise.} \end{cases} \tag{4.9}$$

Integrating A_I (Equation 4.5) from $\theta_I = -\pi/2$ to $\theta_I = \pi/2$ gives

$$V_I = \pi wh - 2\ell(w + h) + \ell^2 \tag{4.10}$$

The selectivity equals $\frac{V}{V_I}$.

These equations assume that the length ℓ of the image line segment is known. It would be convenient to integrate out the length as I did for orientation, but in real images it is not fair to assume that all lengths are equally likely. One possibility is to measure the distribution of image segment lengths over a large set of typical images, and integrate over the distribution. A simpler approach is to measure the average length of an image segment and use the average length for ℓ in the above equations. Alternatively, it may be possible to estimate the percentage, say α, of a model segment that is broken up by the feature detector; this gives $\ell = (1-\alpha)L$ [Grimson91].

4.2.2 Overlapping uncertainty circles

Occasionally the uncertainty circles may overlap, either by intersection or inclusion. The uncertainty circles intersect if $R - r \leq L \leq R + r$. In this case θ_2 is undefined, unless $L = R + r$ (Equation 4.2). Also, ϕ will be undefined if $\ell < R + r - L$ (Equation 4.8), but in this situation the length constraint is reached after $\theta > \pi/2$. To avoid redundancy, we are only interested in orientations of the image segment that are in the range $[0, \pi/2]$. So for convenience we define ϕ to be $\pi/2$ whenever $\ell < R + r - L$. As with the situation of non-overlapping uncertainty circles, there are two cases for the height of the rectangle, depending on whether the orientation of the image segment is less than or greater than $\theta_1 = sin^{-1}\left(\frac{R-r}{L}\right)$ (see Fig. 4-8). In addition, however, there are two cases for the base of the rectangle (Fig. 4-8), depending on whether the orientation is less than or greater than

$$\theta_1' = \pi/2 - \theta_1 = cos^{-1}\left(\frac{R-r}{L}\right) \tag{4.11}$$

There are two basic rules for computing the height and base of the rectangle: (1) When $\theta \leq \theta_1$, use $2r$ for the height; otherwise use $R + r - L\sin\theta$. (2) When $\theta \leq \theta_1'$, use $R + r + L\cos\theta$ for the base; otherwise use $2R$. These two rules lead to four area formulas:

$$\begin{aligned}
a_1 &= (R + r + L\cos\theta - \ell)2r && (4.12)\\
a_2 &= (R + r + L\cos\theta - \ell)(R + r - L\sin\theta) && (4.13)\\
a_3 &= (2R - \ell)2r && (4.14)\\
a_4 &= (2R - \ell)(R + r - L\sin\theta) && (4.15)
\end{aligned}$$

To get the corresponding volume formulas, we need to integrate these formulas over the range of θ. Notice that the first two formulas appear in Equation 4.4; consequently, $v_1(\omega_1, \omega_2)$ and $v_2(\omega_1, \omega_2)$ are given by Equations 4.6 and 4.7, respectively. From Appendix E.2, we have that

$$\begin{aligned}
v_3(\omega_1, \omega_2) &= (2R - \ell)2r(\omega_2 - \omega_1) && (4.16)\\
v_4(\omega_1, \omega_2) &= (2R - \ell)(R + r)(\omega_2 - \omega_1) + (2R - \ell)L(\cos\omega_2 - \cos\omega_1) && (4.17)
\end{aligned}$$

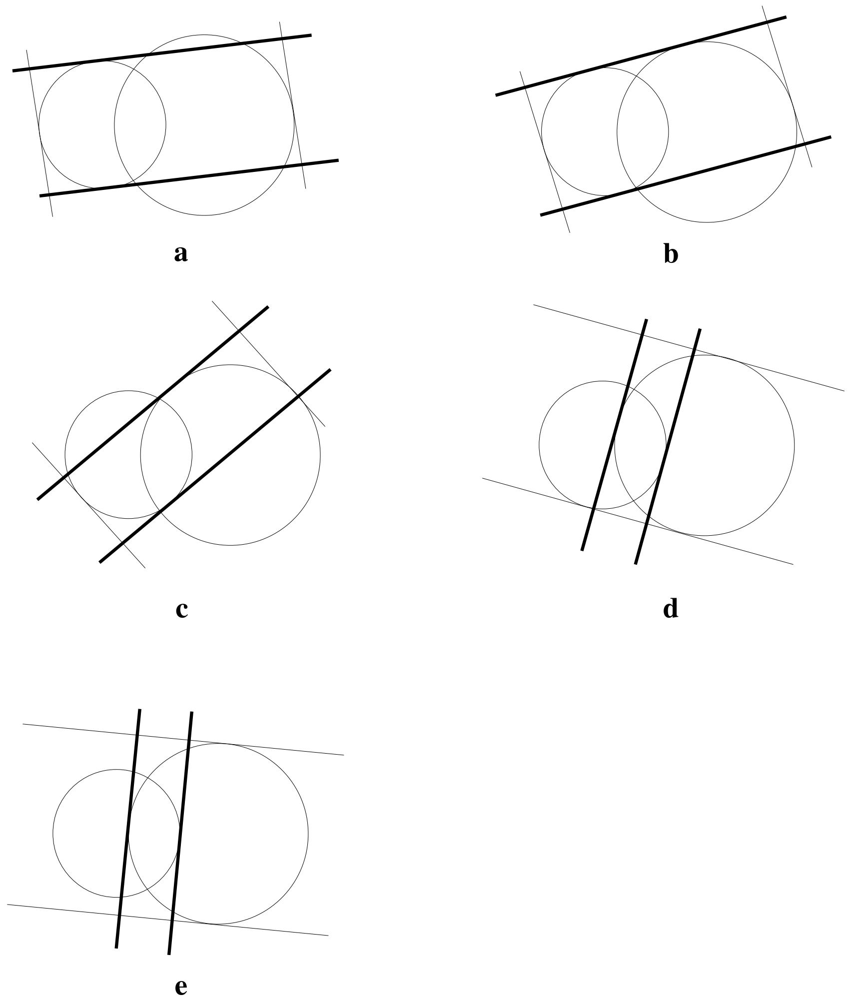

Figure 4-8: Cases where uncertainty circles intersect. The orientation of the image segment increases from picture a to picture e. Let r be the radius of the smaller circle, R be the radius of the larger circle, L be the distance between the center points, θ be the orientation of the image segment, b be the base of the rectangle, and h be the height of the rectangle. Then

a. $\theta \leq \theta_1$,	$b = R + r + L\cos\theta$,	$h = 2r$
b. $\theta = \theta_1$,	$b = R + r + L\cos\theta$,	$h = 2r = R + r - L\sin\theta$
c. $\theta_1 \leq \theta \leq \pi/2 - \theta_1$,	$b = R + r + L\cos\theta$,	$h = R + r - L\sin\theta$
d. $\theta = \pi/2 - \theta_1$,	$b = R + r + L\cos\theta = 2R$,	$h = R + r - L\sin\theta$
e. $\pi/2 - \theta_1 \leq \theta$,	$b = 2R$,	$h = R + r - L\sin\theta$

Rules 1 and 2 apply while $\theta \leq \phi$; otherwise the image segment is too long to fit in the rectangular box. However, this constraint only applies if $\phi \leq \theta'$, because as soon as $\theta \geq \theta'$, the base of the bounding rectangle does not change any more (see cases d and e in Fig. 4-8). Therefore, if $\phi \geq \theta'$, the length of the image segment does not constrain the range of orientations, in which case the maximum orientation of the image segment is $\pi/2$ since the uncertainty circles intersect.

As a final constraint, there exists some volume of translations as long as $\ell \leq R+r+L$, because then the image segment fits in the rectangular box when $\theta = 0$, which is when the orientation of the image segment is the same as the orientation of the model segment. Otherwise the volume of translations is zero.

Putting these constraints together with the volume expressions,

$$V = 2\begin{cases} v_1(0,\phi) & \text{if } \phi \leq \theta_1, \theta_1', \quad l \leq R+r+L, \\ v_1(0,\theta_1) + v_2(\theta_1,\phi) & \text{if } \theta_1 \leq \phi \leq \theta_1', \quad l \leq R+r+L, \\ v_1(0,\theta_1) + v_2(\theta_1,\theta_1') + v_4(\theta_1',\pi/2) & \text{if } \theta_1 \leq \theta_1' \leq \phi, \quad l \leq R+r+L, \\ v_1(0,\theta_1') + v_3(\theta_1',\pi/2) & \text{if } \theta_1' \leq \phi \leq \theta_1, \quad l \leq R+r+L, \\ v_1(0,\theta_1') + v_3(\theta_1',\theta_1) + v_4(\theta_1,\pi/2) & \text{if } \theta_1' \leq \theta_1 \leq \phi, \quad l \leq R+r+L, \\ 0 & \text{otherwise.} \end{cases} \tag{4.18}$$

If the circles overlap but do not intersect, then the smaller circle is contained in the larger, as in Fig. 4-9. In this case, $L \leq R - r$. After shrinking by ℓ along the base, the rectangle in the figure has area $(2R - \ell)2r$. Integrating this expression gives

$$V = \begin{cases} 2\pi r(2R-\ell) & \text{if } \ell \leq 2R, \\ 0 & \text{otherwise.} \end{cases} \tag{4.19}$$

When the uncertainty circles overlap, the selectivities for lines may be larger than for points. This is in part because for lines we did not insist that the endpoints be unoccluded. In addition, when the circles overlap the rectangular upper bound is not as tight an estimate. Since we are using line features to improve on points, we could prevent lines from doing worse by instead using the selectivities of the endpoints whenever their average selectivity is less than the line selectivity. In effect, this insists that the endpoints be unoccluded if the predicted model edge is short enough that the endpoint uncertainty regions overlap.

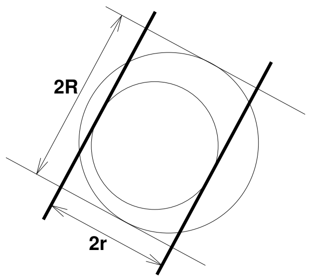

Figure 4-9: Rectangular upper bound for when the smaller uncertainty circle is contained in the larger, or when $L \leq R - r$.

4.2.3 Summary

Given a model line segment, we can compute its selectivity, μ, as follows. Let r and R be the radii of the two uncertainty circles for the endpoints of the line segment, such that $r \leq R$. r and R can be computed using the technique given in Chapter 3 or else, if the models are planar, using the known analytic solution. Next, let L be the distance between the centers of the two circles, and let ℓ be the expected length of a random line segment in the image. Define

$$\begin{aligned}
v_1(\omega_1, \omega_2) &= (R + r - \ell)2r(\omega_2 - \omega_1) + 2rL(\sin \omega_2 - \sin \omega_1) \\
v_2(\omega_1, \omega_2) &= (R + r - \ell)(R + r)(\omega_2 - \omega_1) + (R + r - \ell)L(\cos \omega_2 - \cos \omega_1) \\
&\quad + (R + r)L(\sin \omega_2 - \sin \omega_1) - \frac{1}{2}L^2(\sin^2 \omega_2 - \sin^2 \omega_1) \\
v_3(\omega_1, \omega_2) &= (2R - \ell)2r(\omega_2 - \omega_1) \\
v_4(\omega_1, \omega_2) &= (2R - \ell)(R + r)(\omega_2 - \omega_1) + (2R - \ell)L(\cos \omega_2 - \cos \omega_1)
\end{aligned}$$

If $R + r \leq L$, let

$$\theta_1 = \sin^{-1} \frac{R - r}{L}, \quad \theta_2 = \sin^{-1} \frac{R + r}{L}, \quad \phi = \cos^{-1} \left(\frac{\ell - (R + r)}{L} \right).$$

Otherwise, if $R - r \leq L \leq R - r$, let

$$\theta_1 = \sin^{-1} \frac{R - r}{L}, \quad \theta_1' = \cos^{-1} \frac{R - r}{L}, \quad \phi = \begin{cases} \cos^{-1} \left(\frac{\ell - (R+r)}{L} \right), & \text{if } \ell \geq R + r - L, \\ \pi/2 & \text{otherwise.} \end{cases}$$

Next, if $R + r \leq L$,

$$V = 2 \begin{cases} v_1(0, \phi) & \text{if } \phi \leq \theta_1, \quad l \leq R + r + L, \\ v_1(0, \theta_1) + v_2(\theta_1, \phi) & \text{if } \theta_1 \leq \phi \leq \theta_2, \quad l \leq R + r + L, \\ v_1(0, \theta_1) + v_2(\theta_1, \theta_2) & \text{if } \theta_2 \leq \phi, \quad l \leq R + r + L, \\ 0 & \text{otherwise.} \end{cases}$$

Otherwise if $R - r \leq L \leq R + r$,

$$V = 2 \begin{cases} v_1(0, \phi) & \text{if } \phi \leq \theta_1, \theta_1', \quad l \leq R + r + L, \\ v_1(0, \theta_1) + v_2(\theta_1, \phi) & \text{if } \theta_1 \leq \phi \leq \theta_1', \quad l \leq R + r + L, \\ v_1(0, \theta_1) + v_2(\theta_1, \theta_1') + v_4(\theta_1', \pi/2) & \text{if } \theta_1 \leq \theta_1' \leq \phi, \quad l \leq R + r + L, \\ v_1(0, \theta_1') + v_3(\theta_1', \pi/2) & \text{if } \theta_1' \leq \phi \leq \theta_1, \quad l \leq R + r + L, \\ v_1(0, \theta_1') + v_3(\theta_1', \theta_1) + v_4(\theta_1, \pi/2) & \text{if } \theta_1' \leq \theta_1 \leq \phi, \quad l \leq R + r + L, \\ 0 & \text{otherwise.} \end{cases}$$

Otherwise if $L \leq R - r$,

$$V = \begin{cases} 2\pi r(2R - \ell) & \text{if } \ell \leq 2R, \\ 0 & \text{otherwise.} \end{cases}$$

Finally,

$$V_I = \pi wh - 2\ell(w + h) + \ell^2$$
$$\mu = \frac{V}{V_I}$$

4.3 Expected Selectivities of Line Features

To compare the effect of line segments versus points, the next experiment estimates the expected selectivity of line features for the telephone model. The expected selectivity for random models should be similar.

Experiment 7: Expected selectivity of line features for the telephone model

Method

To compute the expected selectivity, I used the formula given in the last section. I ran a series of the same trials from Experiments 5 and 6 when the selectivity of point features was computed. For each trial, I used each pair of uncertainty circles that corresponds to a line segment in the telephone model (Fig. 3-2) and computed the line segment selectivities. This was repeated for various lengths of the average image line segment and for various amounts of fragmentation, α.

Length	$\overline{\mu}$
5	.001618
10	.001577
20	.001491
30	.001396
40	.001286
50	.001166
60	.001033
70	.0009125
80	.0008085
90	.0007057
100	.0006231

Table 4.1:
Expected selectivities of line features for various lengths of an image segment, using the telephone model.

α	$\overline{\mu}$
0.00	.000647
0.25	.001017
0.50	.001311
0.75	.001550
1.00	.001750

Table 4.2:
Expected selectivities of line features for various amounts of fragmentation, α, using the telephone model.

Results and Discussion

For 1000 trials, the selectivities of 9560 line uncertainty regions were computed and averaged. Tables 4.1 and 4.2 give the results. As expected, the selectivities for lines are much less than for points (compare to Table 3.6). For the telephone, we can see that the largest selectivity using line features, .001750, is less than half the selectivity using point features, .003722.

Chapter 5

Sensitivity to False Positives

There are a number of important questions we would like to answer which depend on the selectivity of a model feature. In particular, given that there is occlusion, what percent of the total length of the model features must be matched in order to keep the probability of a false positive less than some limit? How does this percentage vary with the numbers of model and image features? Also, how many image features can there be before the probability of a false positive exceeds some limit, that is, how much clutter can the system withstand?

Grimson et al. have shown how to use the expected selectivity of the uncertainty regions to answer the above questions [Grimson91] [Grimson92a] [Grimson92b], and so I will apply their analysis here. Let $\overline{\mu}$ be the expected selectivity, let s be the number of unmatched features in the image, let m be the number of unmatched features in the model, and let m' be the number of point features in the model that are used for generating hypotheses. Assuming that the s unmatched image features occur independently and at random, the probability of at least one image feature appearing in a propagated region with selectivity $\overline{\mu}$ is

$$p = 1 - (1 - \overline{\mu})^s \tag{5.1}$$

The probability of exactly k of the m propagated regions having at least one random feature is

$$q_k = \binom{m}{k} p^k (1-p)^{m-k} \tag{5.2}$$

The probability of at least k of the m propagated regions having at least one random feature is

$$w_k = 1 - \sum_{i=0}^{k-1} q_i = 1 - \sum_{i=0}^{k-1} \binom{m}{i} p^i (1-p)^{m-i} \tag{5.3}$$

w_k is the probability of a false positive of size k. If we match a fixed image triple to all possible model triples, the probability that at least one of the matches leads to a false positive of size k is

$$e_k = 1 - (1 - w_k)^{\binom{m'}{3}} \tag{5.4}$$

5.1 Limits on Scene Clutter

A recognition scheme based on extended model features will suffer from false positives if a scene becomes extremely cluttered. It would be useful, then, to know how much clutter a recognition system can accommodate before the probability of a false positive is significant. We can use Equation 5.4 to estimate this limit. To allow for partial occlusion, let f be the fraction of model features that must be matched to keep the probability of a false positive at most δ, where δ is a preset limit. Substituting mf for k, we want to find the maximum s such that $e_{mf} \leq \delta$. This inequality can be solved numerically to get the maximum s.

Table 5.1 shows the results for $\delta = .001$ (the numbers for $\delta = .01$ and $\delta = .0001$ are similar). Real images can easily contain as many as 500 features. The limits for the uncertainty propagation technique of [Grimson92a] are very low. Although the numbers are greatly improved using uncertainty circles, it is only when line segments are used that numbers of features are in the range of images with substantial amounts of scene clutter.

5.2 Accepting a Partial Match

When the extended features of a model are used for verification, we would like to know what percent of the extended features must be matched before we can stop looking for more matches. We can use Equation 5.3 to set a threshold on this percentage such that the chance that a false positive will arise is less than a preset limit. Specifically, given a three-point match, can compute the minimum f such that $w_{mf} \leq \delta_2$, where δ_2 is preset. Table 5.2 shows the results for line segments. For comparison, the recognition system of [Huttenlocher88] used $f = .5$ as a threshold on the percentage of the model to verify;

Method	α	$f = 0.25$	0.50	0.75
Line Uncertainty Regions	0.00	161	537	1200
Line Uncertainty Regions	0.25	102	341	763
Line Uncertainty Regions	0.50	79	265	592
Line Uncertainty Regions	0.75	67	224	500
Line Uncertainty Regions	1.00	59	198	443
Uncertainty Circles	–	31	97	216
Grimson92a	–	15	43	95

Table 5.1:

Approximate limits on the number of sensory features for different amounts of fragmentation α, and for different fractions f of unoccluded model features. Table is for $\epsilon = 5$, $\delta = .001$, for line segments $m = m' = 200$ (line uncertainty regions), and for points $m = 197$ and $m' = 200$ (uncertainty circles and [Grimson92a]).

α	$\delta_2 = .01$	.001	.0001
0.00	.36	.38	.41
0.25	.49	.51	.54
0.50	.57	.60	.62
0.75	.63	.66	.68
1.00	.67	.70	.72

Table 5.2:

Predicted termination thresholds for different amounts of occlusion α, and for different limits δ_2 on the false positive probability. Table is for $\epsilon = 5$, $m = m' = 200$, and $s = 500$.

this agrees with the table when the amount of fragmentation is $\alpha = 25\%$.

5.3 Conclusion

The expected selectivities of model features can be used to estimate two important quantities. The first is a limit on the number of spurious features there can be before the likelihood of false positive becomes significant. With such a limit, we can tell in advance, given a model and an image, whether the recognition system is likely to succeed in finding the model in the image.

The second quantity is a threshold on the percentage of model features to match. Such a threshold can be used actively by a verification system to cut short the search for matches.

Chapter 6

Likelihood of a Hypothesis

In this chapter, I give a criterion which can be used to rank a hypothesis of three matched model and image points, according to how likely it is of being correct. This step is similar in purpose to the quick check used by Huttenlocher at the beginning of his verification stage [Huttenlocher88]. Huttenlocher used simple heuristics to filter hypotheses, whereas here I utilize the uncertainty propagation analysis to rank hypotheses formally, based on a probabilistic model.

At the point where likelihoods are assigned (step 2d of the algorithm of Section 1.3), the alignment system has hypothesized a pairing between three model and image points, and the basic question is whether or not the pairing is correct. To make this determination, the system looks for additional matches to confirm the three suggested ones. Using the hypothesis, the extended model features (points, line segments, segments of curves) are transformed and projected into the image. Then the correct search regions are computed and searched for additional matches. Once the additional matches have been collected, line segments that are nearly collinear and have proximate opposite endpoints should be combined. Also, curve segments should be combined if they appear broken. Given a set of candidate image features for each predicted model feature, we wish to estimate how likely it is that the hypothesis is correct.

6.1 Formula for the Likelihood

To compute the likelihood, assume in general that all features in the image that do not come from the model arise at random. In truth, such features arise from clutter in the scene, occluding objects, and noise; so I am assuming the features these events introduce effectively occur at random. This assumption has been made before for analyzing the verification stage of recognition and has yielded accurate results [Grimson91]. In addition, I assume that none of the uncertainty regions overlap. Let M be the event that the particular matches for the model features were found, and let H be the event that a given three-point match is correct. Then the probability that the matches arose when the model was present is $p(M|H)$. Similarly, the probability that the matches arose at random is the probability that the matches arose when the hypothesis is wrong, which is $p(M|\overline{H})$. However, we are interested in the probability of H given the event M. From Bayes' rule,

$$\begin{aligned} p(H|M) &= \frac{p(M|H)p(H)}{p(M)} = \frac{p(M|H)p(H)}{p(M|H)p(H) + p(M|\overline{H})p(\overline{H})} \\ &= \frac{1}{1 + \frac{p(M|\overline{H})}{p(M|H)}(\frac{1}{p(H)} - 1)} \end{aligned} \tag{6.1}$$

Notice that we also need to compute $p(H)$, the a priori probability that the three point match is correct. Let H_m be the event that the three matched model points are visible, and let H_i be the event that the three matched image points were produced by the three model points. Then $p(H) = p(H_i|H_m)p(H_m)$. If we have information about self occlusion, we may be able to estimate $p(H_m)$ for different triples of model points. Otherwise we can assume that the model is transparent, in which case $p(H_m)$ is the same for triples of the model, and hence equals the probability that the model appears in the image.

As for $p(H_i|H_m)$, this is the probability that the three model features project to within the error bounds of their corresponding image features. We could estimate this off-line for every triple of model features by sampling the viewing sphere and computing the fraction of viewpoints from which the projected model points can be scaled, rotated in 2D, and translated in 2D to lie within the uncertainty regions of the three image points.

Alternatively, we could estimate $p(H_i|H_m)$ at run-time, using the pose-space analysis in [Grimson92a]. More simply, we could assume that for the most part pose space is uniformly distributed. Then $p(H_i|H_m)$ is the probability that any point in pose space gives a triple consistent with the image points. Since there is a point in pose space for every image triple, this probability is $\left(\frac{\pi\epsilon^2}{A_I}\right)^3$, where ϵ is the error bound for the matched image points and A_I is the area of the image.

We still need to determine $\frac{p(M|\overline{H})}{p(M|H)}$. As mentioned, $p(M|\overline{H})$ is the chance the matches occurred at random. Let r equal the number of unmatched image features. Further, let μ_i denote the selectivity of region R_i, and r_i be the number of features found in R_i, for $i = 1, 2, \ldots, k$. (Selectivity was defined in Section 3.5.) Also, let

$$\begin{aligned} r_{k+1} &= r - \sum_{i=1}^{k} r_i && (6.2) \\ \mu_{k+1} &= 1 - \sum_{i=1}^{k} \mu_i && (6.3) \end{aligned}$$

From the assumption that the regions do not intersect, μ_{k+1} is the selectivity of the background. For non-intersecting regions, the chance of r_1 features landing in R_1, r_2 landing in R_2,..., r_k landing in R_k, and r_{k+1} landing in the background is $\mu_1^{r_1}\mu_2^{r_2}\cdots\mu_{k+1}^{r_{k+1}}$. The number of ways to choose $r_1, r_2, \ldots, r_k$ features from r is given by the multinomial coefficient,

$$\binom{r}{r_1, r_2, \ldots, r_{k+1}} = \frac{r!}{r_1! r_2! \cdots r_{k+1}!}\ ,$$

so that

$$p(M|\overline{H}) = \binom{r}{r_1, r_2, \ldots, r_{k+1}} \mu_1^{r_1}\mu_2^{r_2}\cdots\mu_{k+1}^{r_{k+1}} \tag{6.4}$$

Next, assume that if the hypothesis is correct, then the model features the system found matches for were not actually occluded. Then we get $p(M|H)$ by just subtracting one feature from every propagated region:

$$p(M|H) = \binom{r-k}{r_1 - 1, r_2 - 1, \ldots, r_k - 1, r_{k+1}} \mu_1^{r_1-1}\mu_2^{r_2-1}\cdots\mu_k^{r_k-1}\mu_{k+1}^{r_{k+1}} \tag{6.5}$$

Dividing Equation 6.4 by 6.5,

$$\frac{p(M|\overline{H})}{p(M|H)} = \frac{r!}{(r-k)! r_1 r_2 \cdots r_k} \mu_1\mu_2\cdots\mu_k \tag{6.6}$$

6.2 Modified Formula for the Likelihood

Equation 6.1 with Equation 6.6 gives the correct likelihood according to the assumptions, but did those assumptions gives us we want? According to the formula, when an uncertainty region is small relative to the size of the image, the chance that a hypothesis is correct increases as the number of matched features in a region goes up. This is because it is unlikely for more and more features to randomly fall in the same small region. The problem is that it is unclear that this behavior is desired. When there happen to be many features in a region, then there probably exists some image event that violates the assumption that the features arose at random. This event most likely is not due to the object we are seeking, in which case we would not want the probability of the hypothesis to increase with the number of features in the region.

A safer approach is to not use the actual numbers of features in the regions, but only the fact that potential matches exist. For this approach, I re-define M to be the event that matches exist in those same uncertainty regions. As before, we assume that the model features represented by M are not actually occluded, so that $p(M|H) = 1$. By so doing, some hypotheses will be ranked higher than they should. If a threshold is used to take the best-ranking hypotheses, then there simply will be more hypotheses to verify later. With $p(M|H) = 1$, Equation 6.1 becomes

$$p(H|M) = \frac{1}{1 + p(M|\overline{H})(1/p(H) - 1)} \tag{6.7}$$

In this formula, $p(M|\overline{H})$ is the probability of a *random conspiracy*, that is, the probability that at least one random image feature falls in every region represented by M. The likelihood of this happening is the sum of the probabilities of all the ways random features can fall in the regions. For r uniformly-distributed features, the chance that r_1 fall in region R_1, r_2 in R_2,..., r_k in R_k is given by Equation 6.4, which happens to be for the old $p(M|\overline{H})$. As before, let $\mu_{k+1} = 1 - \sum_{i=1}^{k} \mu_i$. Also, define

$$r_{k+1}(r_1, r_2, \ldots, r_k) = r - \sum_{i=1}^{k} r_i.$$

I will abbreviate $r_{k+1}(r_1, r_2, \ldots, r_k)$ by r_{k+1}, but keep in mind that r_{k+1} is a function while μ_{k+1} is constant. Summing over all possible values of the r_i's, for $i = 1, 2, \ldots, k$, the chance of a random conspiracy is

$$p(M|\overline{H}) = \sum_{r_1=1}^{r-k+1} \sum_{r_2=1}^{r-k+2-r_1} \cdots \sum_{r_k=1}^{r-r_1-r_2-\cdots r_{k-1}} \binom{r}{r_1, r_2, \ldots, r_{k+1}} \mu_1^{r_1} \mu_2^{r_2} \cdots \mu_{k+1}^{r_{k+1}} \tag{6.8}$$

This formula involves a large number of computations of the expression from Equation 6.4. The number of computations is exponential in r and k, where r is the number of unmatched image features and k is the number of predicted model features for which potential matches exist. Appendix F derives a recurrence relation that computes one minus the same result. Let $S_1, S_2, \ldots, S_k$ be the "sizes" of the uncertainty regions, and let S_I be the "size" of the image. For points, the sizes are gives by the areas of the uncertainty regions, and for lines the sizes are given by the volumes. The recurrence is,

$$q_r(S_I; S_1, \ldots, S_k) = \begin{cases} Q_r(S_I; S_k)\,(1 - q_r(S_I - S_k; S_1, \ldots, S_{k-1})) + q_r(S_I; S_1, \ldots, S_{k-1}) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } k > 1, \\ Q_r(S_I; S_1) \qquad\qquad\qquad\qquad\qquad\qquad\quad\ \ \text{if } k = 1 \text{ and } S_1 \le S_I, \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ \ \text{otherwise.} \end{cases} \tag{6.9}$$

where

$$Q_r(S_I; S) = \left(1 - \frac{S}{S_I}\right)^r \tag{6.10}$$

This expression has repeated sub-problems at every recursive call, such that only one additional subproblem is generated at each level. At the bottom level, there are k expressions to evaluate, namely $Q(S_I, S_i)$, for $i = 1, \ldots, k$, which are the only times r is used. Dynamic Programming, then, can be used to compute the result of the recurrence in time quadratic in k. Further, since r is the exponent in the equation for $Q_r(S_I; S)$, the time is that needed to compute the power, which is logarithmic in r.

We may ask where this approach is likely to fail. The real trouble for the method is regions where there exist potential matches, but the true feature is either hidden or was not detected. Such regions will give positive evidence for the hypothesis, even though the correct feature is not there. In these cases, I generously assigned $p(M|H) = 1$. As a result, there may be many high-ranked hypotheses instead of a few. This situation seems likely for point features, since spurious points can arise almost anywhere. For extended features, on the other hand, such as line segments and segments of curves, it is much less likely for a long feature to randomly fall in an uncertainty region, and so the chance of the true feature being covered up while random ones appear is expected to be small.

6.3 Summary

The goal of the last two sections was to provide a means for distinguishing a few, most-probable hypotheses, using the extended features of a model. For ease of use, I will summarize the method for line segments. It should be straightforward to apply the method to points, since they are a simpler case. I chose line segments since points are less useful for verification, and line segments are used more commonly.

Given a three-point hypothesis, project the model line segments into the image and compute their line uncertainty regions, making sure to expand out the boundaries by ϵ. In detail, for each endpoint of a model line segment, first compute its uncertainty circle: The center point is at the nominal point and the radius r equals the maximum distance from the nominal point to one of the $8^3 = 512$ sample points, plus ϵ. The line uncertainty region is defined by the uncertainty circles for the endpoints and their common outer tangents (Fig. 4-1). Next, search the uncertainty regions to see which ones have candidate matches. Use the method of Chapter 4 to compute the volumes V of each line uncertainty region and the volume V_I of the image. Also, let s be the total number of image features, let $r = s - 3$, and let

$$p(H) = \left(\frac{\pi\epsilon^2}{wh}\right)^3$$

To use the approach of Section 6.1, next calculate the line selectivities using $\mu = \frac{V}{V_I}$. Then compute

$$\frac{p(M|\overline{H})}{p(M|H)} = \frac{r!}{(r-k)!r_1r_2\cdots r_k}\mu_1\mu_2\cdots\mu_k,$$

from which the likelihood of the hypothesis is

$$p(H|M) = \frac{1}{1 + \frac{p(M|\overline{H})}{p(M|H)}(\frac{1}{p(H)} - 1)}.$$

To use instead the approach of Section 6.2, define the recurrence relation,

$$q_r(S_I; S_1, \ldots, S_k) = \begin{cases} Q_r(S_I; S_k)\,(1 - q_r(S_I - S_k; S_1, \ldots, S_{k-1})) + q_r(S_I; S_1, \ldots, S_{k-1}) & \\ & \text{if } k > 1, \\ Q_r(S_I; S_1) & \text{if } k = 1 \text{ and } S_1 \leq S_I, \\ 0 & \text{otherwise.} \end{cases}$$

with

$$Q_r(S_I; S) = \left(1 - \frac{S}{S_I}\right)^r$$

Using the recurrence, compute $q_r(V_I; V_1, \ldots, V_k)$. Then let

$$p(H|\overline{M}) = 1 - q_r(V_I; V_1, \ldots, V_k)$$

and, finally, the likelihood of the hypothesis is

$$p(H|M) = \frac{1}{1 + p(M|\overline{H})(1/p(H) - 1)}$$

6.4 Precomputing the Likelihoods

For flat models, it is known that the size of the uncertainty region for a predicted model feature does not change with viewpoint, that is, the size does not change as different image points are hypothesized to match the same triple of model points. For solid models, it may be the case that the size of an uncertainty region changes only a little with viewpoint, if the model is not very elongated. In this case, it would be possible to pre-compute the uncertainty regions for each triple of model points. In addition, for each such triple, the likelihoods could be computed in advance for different subsets of the corresponding propagated regions. Then the model triples could be ordered in advance according to how likely they are of having a subset of propagated regions with a high likelihood of being correct. Despite the possibilities, it must first be determined how sensitive are the uncertainty regions for out-of-plane model features to changing viewpoint.

6.5 Discussion

In deciding on a three-point match, the main mechanism we are banking on is that it is unlikely for features to arise at random in the uncertainty regions. And so the more model features for which we find potential matches, the more likely it is that the hypothesis is correct. For the approach of Section 6.1, note that generally the ratio in Equation 6.6 decreases as k, the number of model features for which candidate matches were found, increases. From Equation 6.1, this causes $p(H|M)$ to increase, as is desired.

There is a secondary effect that realizes that finding candidate matches for many model features may not imply that the hypothesis is correct. In particular, if matches are found within uncertainty regions that are very large, then the matches could just as easily have arisen randomly. It is important to know when we are in such a situation, and to reduce our confidence in the hypothesis. This effect depends on the sizes of the uncertainty regions, which depend on which three points from the model are being used in the hypothesis and where the model is being viewed from. For the approach of Section 6.1, larger uncertainty regions cause the μ_i in Equation 6.6 to increase. From Equation 6.1, this causes $p(H|M)$ to decrease, as is desired.

A tertiary effect on the likelihood of a hypothesis is the chance that the three model points projected to their hypothesized corresponding image points. Although it may seem related, this issue is orthogonal to the issue of the effect on the sizes of the uncertainty regions due to where the model is viewed from. To see this, note that if the model triple is equally likely to project to any image triple, it could still be that which image triple it projects to makes a big difference in the sizes of the uncertainty regions. If we suppose the model triple is equally-likely to be seen from any direction, then it may be that some image triples are very unlikely to arise, despite the fact that every image triple is possible. The reason this issue is important is that it may be possible to have a match for several model features that is unlikely to have arisen at random, but at the same time the model triple has almost a zero chance of projecting to its hypothesized image triple. Using $p(H)$, the analysis above gives us a way of trading off these effects.

Chapter 7

Conclusion

This thesis has four main contributions. The first is a geometric understanding of a fundamental problem in computer recognition, namely, the solution for 3D pose from three corresponding points under weak-perspective projection (Chapter 2). A new solution to the problem was given, and the situations where there is no solution and where the solution is unstable were described. In addition, the new solution was put in perspective with previous solutions, and the three most related earlier solutions were presented in detail and compared.

In addition, Chapter 2 showed how the image position of an unmatched model point can be computed efficiently using the solution for 3D pose. In particular, Chapter 2 gave an expression for the fourth point image position that did not involve going through a model-to-image transformation, but instead computed the position directly from the distances between the three matched points. This is important for alignment-style recognition, since the image positions of the unmatched model points are computed many times while searching for the correct pose of the model.

The second major contribution of this thesis is an error analysis of point features for alignment-style recognition of 3D models from 2D images (Chapter 3). The earlier analysis of [Grimson92a] was conservative in its bounds on the propagated uncertainty, and Chapter 3 showed we can do better. In fact, the analysis in Chapter 3 is almost always a solution, which means its bounds are exact, notably except where the 3D pose solution is inherently unstable. Chapter 3 showed pictures of what the true uncertainty regions look like when the bounds are not exact. In these cases, the bounds conservatively overestimate the exact bounds.

Even though the error propagation technique in Chapter 3 is generally accurate, the technique has the disadvantage of being numerical. Nevertheless, so was the only previous error propagation technique. Moreover, for most recognition problems, the time to compute the solution is effectively constant, as though the solution were analytic.

Another contribution of this thesis is a formula for the selectivity of line features (Chapter 4). The selectivity of a feature can be used to infer the expected performance of recognition systems. It can also be used to set a threshold on how much of a model must be identified in an image before the object is recognized (Chapter 5). To date, a selectivity formula for line features has been provided for recognition involving 2D models and 2D data, and for 3D models and 3D data [Grimson91]. The formula derived here is the first for recognition involving 3D models and 2D data.

The fourth major contribution is a formula for the likelihood of a hypothesized three-point match (Chapter 6). The formula applies to point or line features, and relies on their associated uncertainty regions. The formula is intended to be used actively during recognition to quickly filter hypotheses that have little support from the image.

These four contributions tie together well for building a fast and robust alignment system. The uncertainty analysis provides the correct minimal search regions to guarantee that no correct hypotheses are lost, which makes the recognition insensitive to false negatives. Further, the uncertainty regions can be computed quickly using the error propagation technique and the fast solution for the image position of an unmatched model point. Once computed, the uncertainty regions usually are small enough to be searched rapidly for candidate image features. Then, using the likelihood formula, the current hypothesis can be evaluated.

Chapter 8

Future Work

Having theoretically studied the alignment system proposed in Chapter 1, the next step is to build an alignment system that uses just the extended features of a model to select best hypotheses. The system would be based on geometric features, particularly points and line segments. Furthermore, the system would be compared to other hypothesize-and-test techniques that also use 3D models and 2D images, notably [Lowe85] and [Huttenlocher88].

Another worthwhile study would be to build the complete models suggested in Chapter 1. This requires obtaining complete 3D edge maps and extracting extended features from them. Given the edge maps, it would be useful to show that they can be used to reliably verify the presence or absence of the model when the model pose is known up to uncertainty in the data.

Another problem is to discover the correct shapes and distributions of the uncertainty in image features, instead of just bounding them. Jacobs observed that simply adjusting the size of the bounded error threshold makes a big difference in the effectiveness of his grouping system [Jacobs89]. It would be useful, then, to be able to set this threshold automatically. More generally, it is expected that the errors in features differ significantly across images, as well as across a single image, and that they are dependent. It would be of interest to study the feature detection process from image formation on up to see how errors can alter the features of a model.

Lastly, the proposed system expects a minimal amount of grouping to be performed. Grouping is an area that has both received much attention and has much potential for improvement. Many approaches attempt to do grouping with just line segments, but after line segments have been extracted, too much information has been lost. There are many techniques for segmenting images into regions based on intensities [Haralick85], but typically these simply cluster similar intensities and do not take advantage of higher-level shape information at edges. Grouping should be performed using intensity images together with their edges.

Appendix A

Rigid Transform between 3 Corresponding 3D Points

This appendix computes a rigid transform between two sets of three corresponding points using right-handed coordinate systems built separately on each set of three points. A right-handed system is determined by an origin point, $\vec{o}$, and three perpendicular unit vectors, $(\widehat{u}, \widehat{v}, \widehat{w})$. Given three points in space, $\vec{p}_0$, $\vec{p}_1$, $\vec{p}_2$, we can construct a right-handed system as follows: Let $\vec{p}_{01} = \vec{p}_1 - \vec{p}_0$ and $\vec{p}_{02} = \vec{p}_2 - \vec{p}_0$. Then let

$$\begin{aligned} \vec{o} &= \vec{p}_0 \\ \vec{u} &= \vec{p}_{01} \\ \vec{v} &= \vec{p}_{02} - (\vec{p}_{02} \cdot \widehat{p}_{01})\widehat{p}_{01} \\ \vec{w} &= \vec{u} \times \vec{v} \end{aligned}$$

Let $(\vec{o}_1; \widehat{u}_1, \widehat{v}_1, \widehat{w}_1)$ and $(\vec{o}_2; \widehat{u}_2, \widehat{v}_2, \widehat{w}_2)$ be the coordinate systems so defined for the original and camera-centered points, respectively.

Given a coordinate system $(\vec{o}; \widehat{u}, \widehat{v}, \widehat{w})$, a rigid transformation that takes a point in world coordinates to a point in that coordinate system is given by $(\mathbf{R}, \vec{t})$, where

$$\mathbf{R} = [\widehat{u} \quad \widehat{v} \quad \widehat{w}], \qquad \vec{t} = \vec{o}$$

(see for example [Craig55]); the transformed $\vec{p}$ is $\mathbf{R}\vec{p} + \vec{t}$. Then we can bring a point $\vec{p}$

from the original system to the world and then to the camera-centered system using

$$\mathbf{R_2}\left(\mathbf{R_1}^T(\vec{p}-\vec{t}_1)\right)+\vec{t}_2=\mathbf{R_2}\mathbf{R_1}^T\vec{p}+\vec{t}_2-\mathbf{R_2}\mathbf{R_1}^T\vec{t}_1$$

where

$$\begin{aligned}\mathbf{R_1} &= [\hat{u}_1 \quad \hat{v}_1 \quad \hat{w}_1], \qquad & \vec{t}_1 &= \vec{o}_1\\ \mathbf{R_2} &= [\hat{u}_2 \quad \hat{v}_2 \quad \hat{w}_2], \qquad & \vec{t}_2 &= \vec{o}_2.\end{aligned}$$

Consequently a rigid transformation $(\mathbf{R}, \vec{t})$ that aligns the two coordinate systems is

$$\mathbf{R}=\mathbf{R_2}\mathbf{R_1}^T, \qquad \vec{t}=\vec{t}_2-\mathbf{R_2}\mathbf{R_1}^T\vec{t}_1. \tag{A.1}$$

Appendix B

Solving for the Scale Factor

B.1 Biquadratic for the Scale Factor

This appendix shows

$$4(s^2 R_{01}^2 - d_{01}^2)(s^2 R_{02}^2 - d_{02}^2) = \left(s^2(R_{12}^2 - R_{01}^2 - R_{02}^2) - (d_{12}^2 - d_{01}^2 - d_{02}^2)\right)^2 \tag{B.1}$$

is equivalent to a biquadratic in s.

Expanding Equation B.1,

$$\begin{aligned}
&4\left(s^4 R_{01}^2 R_{02}^2 - s^2(R_{01}^2 d_{02}^2 + R_{02}^2 d_{01}^2) + d_{01}^2 d_{02}^2\right) = \\
&\quad s^4(R_{01}^2 + R_{02}^2 - R_{12}^2)^2 - 2s^2(R_{01}^2 + R_{02}^2 - R_{12}^2)(d_{01}^2 + d_{02}^2 - d_{12}^2) \\
&\quad +(d_{01}^2 + d_{02}^2 - d_{12}^2)^2
\end{aligned}$$

$$\begin{aligned}
&s^4\left(4R_{01}^2 R_{02}^2 - (R_{01}^2 + R_{02}^2 - R_{12}^2)^2\right) \\
&\quad -2s^2\left(2R_{01}^2 d_{02}^2 + 2R_{02}^2 d_{01}^2 - (R_{01}^2 + R_{02}^2 - R_{12})(d_{01}^2 + d_{02}^2 - d_{12}^2)\right) \\
&\quad +\left(4d_{01}^2 d_{02}^2 - (d_{01}^2 + d_{02}^2 - d_{12}^2)^2\right) = 0
\end{aligned}$$

$$as^4 - 2bs^2 + c = 0,$$

where

$$
\begin{aligned}
a &= 4R_{01}^2R_{02}^2 - (R_{01}^2 + R_{02}^2 - R_{12}^2)^2 \\
b &= 2R_{01}^2d_{02}^2 + 2R_{02}^2d_{01}^2 - (R_{01}^2 + R_{02}^2 - R_{12}^2)(d_{01}^2 + d_{02}^2 - d_{12}^2) \\
c &= 4d_{01}^2d_{02}^2 - (d_{01}^2 + d_{02}^2 - d_{12}^2)^2.
\end{aligned}
$$

B.2 Two Solutions for Scale

The following lemma completes the proof of Proposition 1:

Lemma : Let f be either $\left(\frac{d_{01}}{R_{01}}\right)^2$ or $\left(\frac{d_{02}}{R_{02}}\right)^2$. Then

$$af^2 - 2bf + c \leq 0. \tag{B.2}$$

Proof:

$$
\begin{aligned}
&af^2 - 2bf + c \\
&= 4(R_{01}R_{02}\sin\phi)^2 f^2 - \\
&\quad 2\left(2(R_{01}^2d_{02}^2 + R_{02}^2d_{01}^2 - 2R_{01}R_{02}d_{01}d_{02}\cos\phi\cos\psi)\right) f + \\
&\quad 4(d_{01}d_{02}\sin\psi)^2, \qquad \text{from Equations 2.18, 2.19, and 2.20} \\
&= 4\left(R_{01}^2R_{02}^2(1-\cos^2\phi)f^2 - \right. \\
&\quad (R_{01}^2d_{02}^2 + R_{02}^2d_{01}^2 - 2R_{01}R_{02}d_{01}d_{02}\cos\phi\cos\psi)f + \\
&\quad \left. d_{01}^2d_{02}^2(1-\cos^2\psi)\right)
\end{aligned}
\tag{B.3}
$$

Suppose that $f = \left(\frac{d_{01}}{R_{01}}\right)^2$. Then B.3 becomes

$$4\left(-\frac{R_{02}^2d_{01}^4}{R_{01}^2}\cos^2\phi + 2\frac{R_{02}d_{01}^3d_{02}}{R_{01}}\cos\phi\cos\psi - d_{01}^2d_{02}^2\cos^2\psi\right)$$

$$= \quad -4R_{02}^2 d_{01}^2 \left(\frac{d_{01}}{R_{01}} \cos\phi - \frac{d_{02}}{R_{02}} \cos\psi \right)^2$$

Suppose instead that $f = \left(\frac{d_{02}}{R_{02}}\right)^2$. Then B.3 becomes

$$\begin{aligned} & 4\left(-\frac{R_{01}^2 d_{02}^4}{R_{02}^2}\cos^2\phi + 2\frac{R_{01} d_{02}^3 d_{01}}{R_{02}}\cos\phi\cos\psi - d_{01}^2 d_{02}^2 \cos^2\psi\right) \\ = & \quad -4R_{01}^2 d_{02}^2 \left(\frac{d_{02}}{R_{02}} \cos\phi - \frac{d_{01}}{R_{01}} \cos\psi \right)^2 \end{aligned}$$

Either way, $af^2 - 2bf + c \leq 0$.
$\square$

B.3 One Solution for Scale

In the "one solution" case, we wish to know when and if $b^2 - ac = 0$ holds. Using the result of Appendix B.5, this means that

$$4(R_{01}d_{02})^4 \left(t^2 - 2\cos(\phi+\psi)t + 1\right)\left(t^2 - 2\cos(\phi-\psi)t + 1\right) = 0.$$

For this to hold, either

$$t^2 - 2\cos(\phi+\psi)t + 1 = 0 \qquad \text{or} \qquad t^2 - 2\cos(\phi-\psi)t + 1 = 0.$$

Solving for t gives

$$t = \cos(\phi+\psi) \pm i\sin(\phi+\psi) \qquad \text{or} \qquad t = \cos(\phi-\psi) \pm i\sin(\phi-\psi), \tag{B.4}$$

where $i = \sqrt{-1}$. Consequently, there are real values of t that make $b^2 - ac = 0$ only if $\sin(\phi+\psi) = 0$ or $\sin(\phi-\psi) = 0$. These situations occur when $\phi = \pm\psi$ and $\phi = \pm\psi + \pi$. Substituting into Equation B.4 gives that $b^2 - ac = 0$ iff both $\phi = \pm\psi$ or $\phi = \pm\psi + \pi$ and $t = 1$, where $t = 1$ is the same as $\frac{d_{01}}{R_{01}} = \frac{d_{02}}{R_{02}}$.

B.4 No Solutions for Scale

This appendix shows that there always exists a solution to the biquadratic by showing that $b^2 - ac \geq 0$. From Appendix B.5,

$$\begin{aligned}
b^2 - ac &= 4(R_{01}d_{02})^4 \left(t^2 - 2\cos(\phi+\psi)t + 1\right)\left(t^2 - 2\cos(\phi-\psi)t + 1\right) \\
&\geq 4(R_{01}d_{02})^4 \left(t^2 - 2t + 1\right)\left(t^2 - 2t + 1\right) \\
&= 4(R_{01}d_{02})^4 (t-1)^4 \\
&\geq 0
\end{aligned}$$

B.5 Simplifying $b^2 - ac$

In this appendix, I derive that

$$b^2 - ac = 4(R_{01}d_{02})^4 \left(t^2 - 2\cos(\phi+\psi)t + 1\right)\left(t^2 - 2\cos(\phi-\psi)t + 1\right), \tag{B.5}$$

where

$$t = \frac{R_{02}d_{01}}{R_{01}d_{02}}.$$

From Equations 2.18, 2.19, and 2.20,

$$\begin{aligned}
a &= 4(R_{01}R_{02}\sin\phi)^2 \\
b &= 2(R_{01}^2 d_{02}^2 + R_{02}^2 d_{01}^2 - 2R_{01}R_{02}d_{01}d_{02}\cos\phi\cos\psi) \\
c &= 4(d_{01}d_{02}\sin\psi)^2
\end{aligned}$$

Then

$$\begin{aligned}
b^2 &= 4(R_{02}^4 d_{01}^4 - 4R_{02}^3 d_{01}^3 R_{01} d_{02}\cos\phi\cos\psi + 2R_{01}^2 R_{02}^2 d_{01}^2 d_{02}^2 + \\
&\quad 4R_{01}^2 R_{02}^2 d_{01}^2 d_{02}^2 \cos^2\phi\cos^2\psi - 4R_{01}^3 d_{02}^3 R_{02} d_{01}\cos\phi\cos\psi + R_{01}^4 d_{02}^4)
\end{aligned}$$

$$ac = 16R_{01}^2 R_{02}^2 d_{01}^2 d_{02}^2 \sin^2\phi\sin^2\psi$$

$$b^2 - ac = 4\left(R_{02}^4 d_{01}^4 - 4R_{02}^3 d_{01}^3 R_{01} d_{02}\cos\phi\cos\psi + \right.$$

$$
\begin{aligned}
& (2 + 4\cos^2\phi\cos^2\psi - 4\sin^2\phi\sin^2\psi)R_{01}^2R_{02}^2d_{01}^2d_{02}^2 - \\
& 4R_{01}^3d_{02}^3R_{02}d_{01}\cos\phi\cos\psi + R_{01}^4d_{02}^4\Big) \\
= \; & 4(R_{01}d_{02})^4\Big(t^4 - 4\cos\phi\cos\psi t^3 + (2 + 4\cos^2\phi\cos^2\psi - 4\sin^2\phi\sin^2\psi)t^2 - \\
& 4\cos\phi\cos\psi t + 1\Big), \qquad \text{where } t = (R_{02}d_{01})/(R_{01}d_{02}) \\
= \; & 4(R_{01}d_{02})^4\Big(t^4 - 2\left(\cos(\phi+\psi) + \cos(\phi-\psi)\right)t^3 + \\
& (2 + 4\cos(\phi+\psi)\cos(\phi-\psi))\,t^2 - 2\left(\cos(\phi+\psi) + \cos(\phi-\psi)\right)t + 1\Big) \\
= \; & 4(R_{01}d_{02})^4\Big(t^2 - 2\cos(\phi+\psi)t + 1\Big)\Big(t^2 - 2\cos(\phi-\psi)t + 1\Big)
\end{aligned}
$$

Appendix C

Generating Random Image and Model Points

This appendix describes how I generated random triples of image points, random triples of model points, and random point models.

C.1 Random Image Triples

Image triples were formed by randomly selecting three 2D locations from an image; the image had dimensions 454×576. I selected image points within a margin of 20 pixels from the boundary. The reason for the margin is that in Experiment 1 (Section 3.2), I discarded propagated uncertainty regions that overlapped the boundary. In order to save time, I used the margin to avoid generating such regions. This basically assumes that image points close to the boundary can be ignored.

In addition to the constraint from the margin, another restriction I applied was to pick image points that were at least 25px apart and at most 250px apart. The minimum distance is used to avoid degenerate point triples, and the maximum distance is used to reflect the expected size of an object found in an image. To get three points that were between 25px and 250px apart, I began by placing the first point at the origin, (0,0). To get a point at most 250px away from the first, the second point was chosen at random from a square centered at the origin of side $2 * 250 + 1 = 501$px. This step was repeated until a point was at least 25px away and at most 250px away from the first point was selected. The third point was repeatedly chosen at random from the same square until a point at least 25px and at most 250px from both of the first two points was selected. This gave an arbitrary triangle within the given distance bounds.

In order to allow the triangle to arise anywhere in the image, the triangle was then randomly translated by putting the first point at a location randomly chosen from within the margin of the image, until a translation was found that left all three points within the margin.

C.2 Random Model Triples

Given a list of model points, which could come from a random model or a true model, first two different points in the list were selected randomly. Then a third point was repeatedly selected at random until a point was found that was non-collinear with the first two. Three points were considered to be collinear if the triangle formed by the three had any angle greater than 175°.

C.3 Random Models

All the generated models had ten points. Ten was chosen because it is a low bound on the number of points in a model, or, equivalently, the number of propagated regions per trial. I wanted a low bound in order to conservatively estimate how well the uncertainty circles fit. A low bound leads to generating more propagated regions with different poses. Trying more poses increases the chance of hitting cases where the uncertainty circles are fit poorly.

To reflect the final appearance of the model in the image, the model points were all chosen to be within 25px and 250px apart. Note that the initial scale of the model is irrelevant, since scale is computed in the pose solution. To get a 3D model, the first point was put at the origin. Then the other model points were selected at random from a cube centered at the origin with side 501px. In addition, each new model point was repeatedly chosen until it was at least 25px and at most 250px from all the current model points. As with scale, the initial translation of the model is arbitrary, since translation is solved for when the pose is computed.

Appendix D

Computing Areas of the True Uncertainty Regions

This appendix describes how the true uncertainty regions are computed from a model and three matched model and image points. First each model point is tested for whether it is in the plane of the three matched model points. If so, its area is computed from the known analytic solution for this case [Jacobs91].

In general, the model points will not lie in the plane of the matched model points. In these cases, the true regions are computed by uniformly sampling twenty-five points along the circle boundaries of the three matched image points. This gives $25^3 = 15625$ samples for each propagated uncertainty region. To obtain the area, first the propagated sample points are written to an image. Then the outer boundary defined by the points in the image is traversed in a four-connected walk. Lastly, all the pixels inside this outer boundary are counted to get the area. Observe that this method can cause the true area to be overestimated because the pixels inside the four-connected boundary can include eight-connected pixels that are not part of the region.

There are two solutions for each pair of model and image triples, which correspond to a reflection about a plane parallel to the image (Chapter 2). In the pose solution of Chapter 2, H_1 and H_2 represent the differences in the z coordinates between the first model point and the second and third model points, respectively; the differences for the reflected solution are therefore $-H_1$ and $-H_2$. To distinguish the two sets of points corresponding to the two weak-perspective solutions, I use the nominal values of H_1 and H_2, which occur when the matched image points are at their nominal locations. If the nominal H_1 is larger, I take all the solutions with the same sign for H_1 as being from the same region. I do the opposite if the nominal H_2 is larger. For the most part, this method works to separate the two regions as long as they do not overlap. If the propagated regions do overlap, there really is one region, and this method will cause it to split.

Appendix E

Areas and Volumes of Line Uncertainty Regions

E.1 True Area of a Line Uncertainty Region

Given a line segment of known orientation and length, the area of the uncertainty region in Fig. 4-2 can be computed by moving the line segment perpendicular to its orientation. This is shown in Fig. 4-5 parametrized by u. This section computes the uncertainty region area. For simplicity, I assume the uncertainty circles for the endpoints do not intersect.

For a given offset u, we are interested in the distance between the outer intersection points of the line and the circles. From the figure, this distance equals

$$\| (x_1, y_1) - (x_2, y_2) \| = \frac{x_2 - x_1}{\cos\theta} = \frac{y_2 - y_1}{\sin\theta}.$$

Putting the origin at the smaller circle, the equations of the line and circles are

$$\begin{aligned} -x\sin\theta + y\cos\theta &= u & \text{(E.1)} \\ x^2 + y^2 &= r^2 & \text{(E.2)} \\ (x - L)^2 + y^2 &= R^2 & \text{(E.3)} \end{aligned}$$

Assuming $\cos\theta \neq 0$, we can solve for y in Equation E.1 and substitute into Equation E.2:

$$\begin{aligned} x^2 + \left(\frac{u + x\sin\theta}{\cos\theta}\right)^2 &= r^2 \\ \implies\ & x^2\cos^2\theta + (u + x\sin\theta)^2 - r^2\cos^2\theta = 0 \\ \implies\ & x^2 + 2u\sin\theta\, x + u^2 - r^2\cos^2\theta = 0 \\ \implies\ & x = -u\sin\theta \pm \sqrt{r^2 - u^2}\ \cos\theta \\ \implies\ & x_1 = -u\sin\theta - \sqrt{r^2 - u^2}\ \cos\theta, \qquad \text{from Fig. 4-5.} \end{aligned}$$

Note that the discriminant is non-negative, since $|u| \leq r$ from Fig. 4-5. Next, substitute for y from Equation E.1 into Equation E.3:

$$\begin{aligned} (x - L)^2 + \left(\frac{u + x\sin\theta}{\cos\theta}\right)^2 &= R^2 \\ \implies\ & (x - L)^2\cos^2\theta + (u + x\sin\theta)^2 - R^2\cos^2\theta = 0 \\ \implies\ & x^2 + 2(-L\cos^2\theta + u\sin\theta)x + u^2 + L^2\cos^2\theta - R^2\cos^2\theta = 0 \\ \implies\ & x = L\cos^2\theta - u\sin\theta \pm \sqrt{R^2 - (L\sin\theta + u)^2}\ \cos\theta \\ \implies\ & x_2 = L\cos^2\theta - u\sin\theta + \sqrt{R^2 - (L\sin\theta + u)^2}\ \cos\theta, \qquad \text{from Fig. 4-5.} \end{aligned}$$

Again, the discriminant is non-negative: The maximum value of $u = \min(r, R - L\sin\theta)$ (see Figs. 4-5 and 4-6), and so

$$\begin{aligned} u \leq R - L\sin\theta \implies\ & u + L\sin\theta \leq R \\ \implies\ & (u + L\sin\theta)^2 \leq R^2, \qquad \text{since } |u| \leq R. \end{aligned}$$

Given x_1 and x_2,

$$\frac{x_2 - x_1}{\cos\theta} = L\cos\theta + \sqrt{R^2 - (u + L\sin\theta)^2} - \sqrt{r^2 - u^2} \tag{E.4}$$

The area A of the shaded region in Fig. 4-2 equals the integral of Equation E.4 from $u = -r$ to $u = \min(r, R - L\sin\theta)$, if the region exists. The region exists if the image segment's orientation is within the bounds of the line uncertainty region, that is, if

$L \sin\theta \le R + r$ (Fig. 4-4). Thus

$$A = \begin{cases} \int_{-r}^{\min(r, R - L\sin\theta)} \left(L\cos\theta + \sqrt{R^2 - (u + L\sin\theta)^2} - \sqrt{r^2 - u^2} \right) du & \\ & \text{if } L\sin\theta \le R + r, \\ 0 & \text{otherwise.} \end{cases} \tag{E.5}$$

This is not the area we are interested in, however. Instead, we want this area shrunk by the length of the image segment. Let ℓ be the length of the image segment. To compute the desired area, subtract ℓ from the term being integrated, Equation E.4. In addition, we must change the upper limit of the integration, since it is constrained by ℓ. In particular, we need to know if and where term being integrated crosses zero, which is where

$$-\ell + L\cos\theta + \sqrt{R^2 - (u + L\sin\theta)^2} - \sqrt{r^2 - u^2} = 0. \tag{E.6}$$

This equation leads to a quadratic in u:

$$(k_1^2 + 1)u^2 + 2k_1k_2u + k_2^2 - r^2 = 0 \tag{E.7}$$

where

$$k_1 = \frac{L\sin\theta}{L\cos\theta - \ell} \tag{E.8}$$

$$k_2 = \frac{\ell^2 + L^2 - R^2 + r^2 - 2\ell L\cos\theta}{2(L\cos\theta - \ell)} \tag{E.9}$$

Then

$$u = \frac{-k_1k_2 \pm \sqrt{(k_1^2 + 1)r^2 - k_2^2}}{k_1^2 + 1} \tag{E.10}$$

The term being integrated crosses zero if the discriminant is non-negative. There are two solutions because squaring was used in the algebra to obtain Equation E.7 from Equation E.6. If the discriminant is non-negative, let u^* be the u from Equation E.10 that satisfies Equation E.6, and let $u_{\max} = \min(u^*, r, R + r - L\sin\theta)$; otherwise let $u_{\max} = \min(r, R + r - L\sin\theta)$. Then the area of translations is

$$A = \begin{cases} \int_{-r}^{u_{\max}} \left(-\ell + L\cos\theta + \sqrt{R^2 - (u + L\sin\theta)^2} - \sqrt{r^2 - u^2} \right) du & \\ & \text{if } L\sin\theta \le R + r, \\ 0 & \text{otherwise.} \end{cases} \tag{E.11}$$

E.2 Integrating Areas to Volumes

$$\begin{aligned} v_1 &= \int (R + r + L\cos\theta - \ell)2r\,d\theta = \int (R + r - \ell)2r\,d\theta + \int 2rL\cos\theta\,d\theta \\ &= (R + r - \ell)2r\theta + 2rL\sin\theta \end{aligned} \tag{E.12}$$

$$\begin{aligned} v_2 &= \int (R + r + L\cos\theta - \ell)(R + r - L\sin\theta)\,d\theta \\ &= \int (R + r - \ell)(R + r)\,d\theta - \int (R + r - \ell)L\sin\theta + \int (R + r)L\cos\theta\,d\theta \\ &\quad - \int L^2\cos\theta\sin\theta\,d\theta \\ &= (R + r - \ell)(R + r)\theta + (R + r - \ell)L\cos\theta + (R + r)L\sin\theta - \frac{1}{2}L^2\sin^2\theta \end{aligned} \tag{E.13}$$

$$v_3 = \int (2R - \ell)2r\,d\theta = (2R - \ell)2r\theta \tag{E.14}$$

$$\begin{aligned} v_4 &= \int (2R - \ell)(R + r - L\sin\theta)\,d\theta = \int (2R - \ell)(R + r)\,d\theta - \int (2R - \ell)L\sin\theta\,d\theta \\ &= (2R - \ell)(R + r)\theta + (2R - \ell)L\cos\theta \end{aligned} \tag{E.15}$$

Appendix F

Recurrence Relation for the Likelihood of a Hypothesis

Let N_i be the event that none of the uniformly distributed image features landed in region R_i. Then the probability that at least one image feature landed in every region is

$$p(M|\overline{H}) = p(\overline{N_1} \wedge \cdots \wedge \overline{N_k}),$$

which implies

$$\begin{aligned} p(\overline{M}|\overline{H}) &= p(N_1 \vee \cdots \vee N_k) \\ &= p(N_1 \vee \cdots \vee N_{k-1}) + p(N_k) - p\left((N_1 \vee \cdots \vee N_{k-1}) \wedge N_k\right) \\ &= p(N_1 \vee \cdots \vee N_{k-1}) + p(N_k)\left(1 - p(N_1 \vee \cdots \vee N_{k-1}|N_k)\right) \end{aligned} \tag{F.1}$$

$p(N_1 \vee \cdots \vee N_k)$ is a function of the uncertainty region sizes, S_i, for $i = 1, 2, \ldots, k$, the maximum size, S_I, and the number of uniformly distributed features, r. For points, the sizes are gives by the areas of the uncertainty regions, and for lines the sizes are given by the volumes. To make the dependency explicit, define

$$q_r(S_I; S_1, \ldots, S_k) \stackrel{\text{def}}{=} p(N_1 \vee \cdots \vee N_k)$$

Therefore in Equation F.1, $p(N_1 \vee \cdots \vee N_{k-1}) = q_r(S_I; S_1, \ldots, S_{k-1})$.

Next, let us consider $p(N_1 \vee \cdots \vee N_{k-1}|N_k)$. If event N_k occurs, that is, if no features land in the kth region, then all of the features are distributed over the rest of the image, so that

$$p(N_1 \vee \cdots \vee N_{k-1} | N_k) = q_r(S_I - S_k; S_1, \ldots, S_{k-1})$$

Lastly, $p(N_k)$ is the probability that all the features missed the kth region, which equals $(1 - \frac{S}{S_I})^r$. Define

$$Q_r(S_I, S_k) \stackrel{\text{def}}{=} p(N_k).$$

Plugging into Equation F.1, $p(\overline{M}|\overline{H})$ is given by $q_r(S_I; S_1, \ldots, S_k)$, which is determined by the recurrence relation,

$$q_r(S_I; S_1, \ldots, S_k) = \begin{cases} Q_r(S_I; S_k)\,(1 - q_r(S_I - S_k; S_1, \ldots, S_{k-1})) + q_r(S_I; S_1, \ldots, S_{k-1}) & \\ & \text{if } k > 1, \\ Q_r(S_I; S_1) & \text{if } k = 1 \text{ and } S_1 \leq S_I, \\ 0 & \text{otherwise.} \end{cases}$$

where

$$Q_r(S_I; S) = \left(1 - \frac{S}{S_I}\right)^r$$

Bibliography

[1] [Ayache86]
Ayache, N., and O. D. Faugeras, "HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, no. 1, pp. 44-54, January 1986.

[2] [Baird85]
Baird, H. S., *Model-Based Image Matching Using Location*, Cambridge, MA: MIT Press, 1985.

[3] [Ballard81]
Ballard, D. H., "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111-122, 1981.

[4] [Basri88]
Basri, R., and S. Ullman, "The Alignment of Objects with Smooth Surfaces," in *Proc. Second Inter. Conf. Computer Vision*, pp. 482-488, 1988.

[5] [Biederman85]
Biederman, I., "Human Image Understanding: Recent Research and a Theory," *Computer Vision, Graphics, and Image Proc.*, vol. 32, pp. 29-73, 1985.

[6] [Bolles82]
Bolles, R. C., and R. A. Cain, "Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method," *Inter. J. Rob. Res.*, vol. 1, no. 3, pp. 57-82, 1982.

[7] [Bolles83]
Bolles, R. C., P. Horaud, and M. J. Hannah, "3DPO: A Three-Dimensional Part Orientation System," in *Proc. 8th Inter. Joint Conf. Artif. Intell.*, pp. 1116-1120, 1983.

[8] [Brooks81]
Brooks, R. A., "Symbolic Reasoning Among 3-D Models and 2-D Images," *Artificial Intell.*, vol. 17, pp. 285-348, 1981.

[9] [Cass90]
Cass, T. A., "Feature Matching for Object Localization in the Presence of Uncertainty," in *Proc. Third Inter. Conf. Computer Vision*, pp. 360-364, 1990.

[10] [Clark79]
Clark, C. S., W. O. Eckhardt, C. A. McNary, R. Nevatia, K. E. Olin, and E. M. VanOrden, "High-accuracy Model Matching for Scenes Containing Man-Made Structures," in *Proc. Symp. Digital Processing of Aerial Images*, SPIE, vol. 186, pp. 54-62, 1979.

[11] [Costa90]
Costa, M., R.M. Haralick, and L.G. Shapiro, "Optimal Affine-Invariant Point Matching," in *Proc. 6th Israel Conf. on Artificial Intell.*, pp. 35–61, 1990.

[12] [Craig55]
Craig, J. J., *Introduction to Robotics*, Reading, MA: Addison-Wesley, 1955.

[13] [Cyganski85]
Cyganski, D., and J. Orr, "Applications of Tensor Theory to Object Recognition and Orientation Determination," *IEEE Trans. Pat. Anal. Machine Intell.*, vol. 7, no. 6, November 1985.

[14] [Cyganski88]
Cyganski, D., and J. Orr, "Object Recognition and Orientation Determination by Tensor Methods," Advances in Computer Vision and Image Processing, vol. 3, Thomas Huang, Ed., Jai Press, Inc., 1988.

[15] [Ellis87]
Ellis, R. E., "Acquiring Tactile Data for the Recognition of Planar Objects," in *Proc. IEEE Conf. Rob. Aut.*, pp. 1799-1805, 1987.

[16] [Fischler81]
Fischler, M. A., and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, pp. 381-396, June 1981.

[17] [Goad83]
Goad, C. A., "Special Purpose Automatic Programming for 3D Model-Based Vision," in *Proc. DARPA Image Understanding Workshop*, pp. 94-104, 1983.

[18] [Graustein30]
Graustein, W. C., *Introduction to Higher Geometry*, New York: The Macmillan Company, 1930.

[19] [Grimson90a]
Grimson, W. E. L., "The Combinatorics of Object Recognition in Cluttered Environments Using Constrained Search," *Artificial Intelligence*, vol. 44, pp. 121-165, 1990.

[20] [Grimson90b]
Grimson, W. E. L., and D. P. Huttenlocher, "On the Sensitivity of the Hough Transform for Object Recognition," *IEEE Trans. Pat. Anal. Machine Intell.*, vol. 12, no. 3, March 1990.

[21] [Grimson91]
Grimson, W. E. L., and D. P. Huttenlocher, "On the Verification of Hypothesized Matches in Model-Based Recognition," *IEEE Trans. Pat. Anal. Machine Intell.*, vol. 13, no. 12, December 1991.

[22] [Grimson92a]
Grimson, W. E. L., D. P. Huttenlocher, & T. D. Alter, "Recognizing 3D Objects from 2D Images: An Error Analysis," in *Proc. IEEE Conf. Computer Vision Pat. Rec.*, pp. 316-321, 1992.

[23] [Grimson92b]
Grimson, W. E. L., D. P. Huttenlocher, and D. W. Jacobs, "A Study of Affine Matching with Bounded Sensor Error," in *Proc. Second European Conf. Computer Vision*, pp. 291-306, May, 1992.

[24] [Grimson84]
Grimson, W. E. L., and T. Lozano-Pérez, "Model-Based Recognition and Localization from Sparse Range or Tactile Data," *Int. J. Robotics Res.*, vol. 3, no. 3, pp. 3-35, 1984.

[25] [Grimson87]
Grimson, W. E. L., and T. Lozano-Pérez, "Localizing Overlapping Parts by Searching the Interpretation Tree," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, no. 4, pp. 469-482, July 1987.

[26] [Haralick91]
Haralick, R. M., C. Lee, K. Ottenberg, and M. Nolle, "Analysis and Solutions of the Three Point Perspective Pose Estimation Problem," in *Proc. IEEE Conf. Computer Vision Pat. Rec.*, pp. 592-598, 1991.

[27] [Haralick85]
Haralick, R. M., and L. G. Shapiro, "Image Segmentation Techniques," *Computer Vision, Graphics, and Image Proc.*, vol. 29, pp. 100-132, 1985.

[28] [Horn86]
Horn, B. K. P., "Closed-Form Solution of Absolute Orientation Using Unit Quaternions," J. Opt. Soc. Am., vol. 4, no. 4, April 1987.

[29] [Horaud87]
Horaud, R., "New Methods for Matching 3-D Objects with Single Perspective Views," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, no. 3, pp. 401-412, May 1987.

[30] [Horaud90]
Horaud, R., F. Veillon, and T. Skordas, "Finding Geometric and Relational Structures in an Image," in *Proc. First European Conf. Computer Vision*, pp. 374-384, April, 1990.

[31] [Huttenlocher88]
Huttenlocher, D. P., "Three-Dimensional Recognition of Solid Objects from a Two-Dimensional Image," MIT TR 1045, 1988.

[32] [Huttenlocher87]
Huttenlocher, D. P., and S. Ullman, "Object Recognition Using Alignment," in *Proc. First Inter. Conf. Computer Vision*, pp. 102-111, June 1987.

[33] [Huttenlocher90]
Huttenlocher, D. P., and S. Ullman, "Recognizing Solid Objects by Alignment with an Image," *Inter. J. Computer Vision*, vol. 5, no. 2, pp. 195-212, 1990.

[34] [Jacobs87]
Jacobs, D. W., "GROPER: A Grouping Based Recognition System for Two Dimensional Objects," in *Proc. IEEE Workshop on Computer Vision*, 1987.

[35] [Jacobs89]
Jacobs, D. W., "Grouping for Recognition," MIT A.I. Memo 1177, November 1989.

[36] [Jacobs91]
Jacobs, D. W., "Optimal Matching of Planar Models in 3D Scenes," in *Proc. IEEE Conf. Computer Vision Pat. Rec.*, 1991.

[37] [Jacobs92]
Jacobs, D. W., "Recognizing 3-D Objects Using 2-D Images," Ph. D. Thesis, MIT Dept. Elec. Eng. and Computer Sci., 1992.

[38] [Kanade83]
Kanade, T., and J. R. Kender, "Mapping Image Properties into Shape Constraints: Skew Symmetry, Affine-Transformable Patterns, and the Shape-from-Texture Paradigm," Human and Machine Vision, Beck, Hope, and Rosenfeld, Eds., Academic Press, 1983.

[39] [Lamdan88a]
Lamdan, Y., J. T. Schwartz, and H. J. Wolfson, "Object Recognition by Affine Invariant Matching" in *Proc. IEEE Conf. Computer Vision Pat. Rec.*, pp. 335-344, 1988.

[40] [Lamdan88b]
Lamdan, Y., and H. J. Wolfson, "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme," in *Proc. IEEE Conf. Computer Vision Pat. Rec.*, pp. 238-249, 1988.

[41] [Lamdan91]
Lamdan, Y., and H. J. Wolfson, "On the Error Analysis of 'Geometric Hashing'," in *Proc. IEEE Conf. Computer Vision Pat. Rec.*, pp. 22-27, 1991.

[42] [Linnainaa88]
Linnainmaa, S., D. Harwood, and L. S. Davis, "Pose Determination of a Three-Dimensional Object Using Triangle Pairs," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 5, pp. 634-647, Sept. 1988.

[43] [Lowe85]
Lowe, D., *Perceptual Organization and Visual Recognition*, The Netherlands: Kluwer Academic Publishers, 1985.

[44] [Lozano-Pérez87]
Lozano-Pérez, T., "A Simple Motion-Planning Algorithm for General Robot Manipulators," *IEEE J. Rob. Aut.*, vol. 3, no. 3, June 1987.

[45] [Mohan88]
Mohan, R., and R. Nevatia, "Perceptual Grouping for the Detection and Description of Structures in Aerial Images," in *Proc. DARPA Image Understanding Workshop*, pp. 512-526, 1988.

[46] [Reeves89]
Reeves, A. P., and R. W. Taylor, "Identification of Three-Dimensional Objects Using Range Information," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 4, pp. 403-410, April 1989.

[47] [Rigoutsos91]
Rigoutsos, I., and R. Hummel, "Robust Similarity Invariant Matching in the Presence of Noise," in *Eighth Israeli Conf. on Artificial Intell. and Computer Vision*, Tel Aviv, 1991.

[48] [Roberts65]
Roberts, L. G., "Machine Perception of Three-Dimensional Solids," Optical and Electro-Optical Information Processing, J. T. Tippett et al., Eds., MIT Press, Cambridge, MA, 1965.

[49] [Thompson87]
Thompson, D. W., and J. L. Mundy, "Three-Dimensional Model Matching from an Unconstrained Viewpoint," in *Proc. IEEE Conf. Rob. Aut.*, pp. 208-220, 1987.

[50] [Turney85]
Turney, J. L., T. N. Mudge, and R. A. Voltz, "Recognizing Partially Occluded Parts," *IEEE Trans. Pat. Anal. Machine Intell.*, vol. 7, no. 4, July 1985.

[51] [Ullman83]
Ullman, S., "Computational Studies in the Interpretation of Structure and Motion: Summary and Extension," Human and Machine Vision, Beck and Rosenfeld, Eds., Academic Press, 1983.

[52] [Ullman86]
Ullman, S., "An Approach to Object Recognition: Aligning Pictorial Descriptions," MIT A.I. Memo 931, December 1985.

[53] [Ullman89]
Ullman, S., "Aligning Pictorial Descriptions: An Approach to Object Recognition," *Cognition*, vol. 32, no. 3, pp. 193-254, August 1989.

[54] [Ullman91]
Ullman, S., and R. Basri, "Recognition by Linear Combinations of Models," *IEEE Trans. Pat. Anal. Machine Intell.*, vol. 13, no. 10, pp. 992-1006, October 1991.