Robust Affine Structure Matching for 3D Object Recognition

Todd A. Cass
Xerox Palo Alto Research Center
October 9, 1995, V1.0

Abstract

This paper considers a model-based approach to identifying and locating known 3D objects from 2D images. The method is based on geometric feature matching of the model and image data, where both are represented in terms of *local geometric features*. The problems of finding feature correspondences in the presence of geometric uncertainty, missing, and spurious data features are explicitly handled. The method works in *transformation space*, as opposed to methods which work in feature *correspondence space*, and is based on constraint analysis using techniques from computational geometry. In it's most general form the method provides an approach to feature matching which is robust to image feature uncertainty and guarantees finding all feasible matchings in polynomial time. While interesting theoretically, this general approach is impractical because of the computation required.

This paper extends and refines previous work on feature matching using transformation constraint methods by detailing the case of full 3D objects represented as point features and developing geometric algorithms based on *conservative approximations* to the general algorithm which are much more computationally feasible. An interesting contribution of this paper is that the formulation developed makes clear the relationship between transformation constraint methods and the popular alignment methods for feature matching, as well as the relationship of feature matching to affine model indexing.

Keywords: *Object Recognition, Geometric Feature Matching, Pose Recovery, Pose Refinement, Affine Structure, Model Indexing, Computational Geometry.*

1 Introduction

This paper considers the task of identifying and locating known 3D objects in 2D images of a scene containing them by matching geometric representations of a model of the object to geometric representations of the image data—a task often called *model-based object recognition*. We take a *model-based* approach to *interpreting* subsets of sensory data corresponding to *instances* of objects in the scene in terms of *models* of the objects known *a priori*. Both the model and the image data are represented in terms of sets of local geometric features[1] consisting of geometric elements like points, line segments, and curve normals. An approach based on geometric feature matching is used to interpret the image by finding both the model to data feature correspondences and a 3D pose of the model which could produce the observed instance.

Geometric feature matching is difficult because the correspondences between the model features and the corresponding subset of data features are unknown, there is clutter from data features with no model correspondence, there are model features which have no corresponding data feature due

[1] Throughout this paper we will follow the convention that terms being effectively defined will appear in sans-serif bold, while items being emphasized for clarity will appear in *italics*.

to occlusion and other processes, and there is geometric uncertainty in the measurement of the data features arising from the imaging process, the feature representation process, and scene events such as occlusion.

This paper greatly extends and refines previously reported techniques for geometric matching based on discrete constraints and methods from computational geometry first presented for the case of 2D objects and planar transformations[9, 10, 12], and 2D objects and 3D transformations[13, 14]. This paper considers the case of 3D objects and 3D transformations.

1.1 Contributions

This paper presents a technique for feature matching based on analyzing constraints on feasible feature matchings in transformation space. It applies to affine feature matching of 2D and 3D point sets and 2D and 3D affine motion. The main contribution is a method for carefully projecting the constraints in the high-dimensional transformation space, an 8-dimensional space for 3D affine motion, down to a two dimensional transformation space. This approximation is conservative in that it will never exclude feasible matches. This approximate projection then allows the use of planar geometric algorithms (which is very important due to the great difficulty of geometric computations in higher dimensions) for analyzing these transformation constraints to find feasible matchings. Efficient algorithms for this planar geometric analysis are presented, and extensive experimental results of their application to matching are presented. Additionally the formalism developed makes clear the relationship between this approach to feature matching and affine model indexing and structure from motion[28].

2 Formalizing the Geometric Feature Matching Problem

2.1 An Abstract Formulation

This section gives a summary of this geometric feature matching paradigm; a more detailed explanation can be found in previous papers[13, 12]. Two important criteria by which to analyze methods for model matching, which are satisfied by the most general form of our approach, are robustness to geometric uncertainty in the data measurements so no correct feature correspondences are missed; and completeness so that *all* sets of geometrically consistent feature correspondences are found, even if there are one, many, or no model instances in the image. If there were no uncertainty a provably correct solution to the matching problem could be achieved by a polynomial-time *alignment* procedure[25].

The local geometric features used here consist of points with associated orientations. Denote a model feature $\mathbf{m} = (\mathbf{m}^p, \mathbf{m}^\theta)$ by an ordered pair of vectors representing the feature's position and orientation, respectively; denote the measured geometry of a data feature similarly by $\mathbf{d} = (\mathbf{d}^p, \mathbf{d}^\theta)$. Central to this formulation is the use of a quantized uncertainty distribution model of uncertainty[13, 8] where the distribution of the deviation of the true value of a geometric quantity from its measured value in the image is represented by discrete quantized regions—in the simplest case a uniform uncertainty bound which we use here[22, 5]. The true value of a geometric parameter is assumed to lie within a bounded neighborhood of its measured value; denote the bounded

uncertainty regions U_p^i and U_θ^i for the data feature position and orientation measurements, respectively, for feature $\tilde{\mathbf{d}}_i$. We let $\tilde{\mathbf{d}}_i$ denote the measured value of a model feature and $\mathbf{d}_i$ denote the value without error. Assume that the uncertainty in position and orientation are independent. The true position of $\mathbf{d}_i$ falls in the set U_p^i, and its true orientation falls in the set U_θ^i.

There have been many geometric feature matching techniques for object recognition explored in the literature (see [35, 6, 20, 5, 22, 25] for a sampling). For clarity we will describe our formulation. The essential idea of local-feature geometric matching is to transform an object model representation so that it is geometrically similar, feature by feature, to the representation of an instance in the image of the object in a scene. Define a match as a pairing of a model feature and a data feature, and a match set, denoted $\mathcal{M}$, as an arbitrary set of matches—some subset of all possible matches. The transformed model feature set is geometrically similar to a data feature set when the transformation mapping the model into the image data geometrically *aligns* a substantial fraction of individual model and data feature matches to within the uncertainty bounds. A transformed model feature is aligned with a data feature when it falls within the geometric uncertainty region for the data feature. For example, if T_p and T_θ together represent a transformation on a model feature $\mathbf{m}_i$, $T[\mathbf{m}_i] \equiv (T_p[\mathbf{m}_i^p], T_\theta[\mathbf{m}_i^\theta])$, then the features $(\mathbf{m}_i, \tilde{\mathbf{d}}_j)$ are aligned by T if $T_p[\mathbf{m}_i^p] \in U_p^j$ and $T_\theta[\mathbf{m}_i^\theta] \in U_\theta^j$. The set of transformation T geometrically feasible for a particular model and data feature match $(\mathbf{m}_i, \tilde{\mathbf{d}}_j)$ is the set of transformations $\mathcal{F}^{ij} = \{T \in \Omega | T[\mathbf{m}_i] \in U^j\}$. A match set is called geometrically consistent if and only if $\bigcap_{(i,j)} \mathcal{F}^{ij} \neq \emptyset$ for (i,j) such that $(\mathbf{m}_i, \mathbf{d}_j) \in \mathcal{M}$, that is, exactly when there exists some transformation which is feasible for all matches $(\mathbf{m}_i, \mathbf{d}_j) \in \mathcal{M}$.

The transformation constraints which form the set $\mathcal{F}^{ij}$ of geometric feasibility for a particular individual model and data feature match $(\mathbf{m}_i, \tilde{\mathbf{d}}_j)$ form the core elements of the geometric matching method we use. For example, in the cases considered in this paper, $\mathcal{F}^{ij}$ is a polytope formed by the intersection of halfspaces. The constraint surfaces forming the boundaries of these polytopes form an *arrangement* (in the computational geometric sense) which partitions the transformation parameter space into equivalence classes. The key insight of this approach is that because of this constraint arrangement there are a polynomial number of *equivalence classes* of transformations aligning subsets of the model feature set with subsets of the data feature set. Finding these *transformation equivalence classes* is equivalent to finding all qualitatively different geometrically feasible match sets, and thus *all* feasible interpretations of the data in terms of the model. This requires time polynomial in the size of the feature sets. Given a model of uncertainty this can be an exact computation: we need not search through an exponential number of sets of possible feature correspondences, consider an infinite set of possible transformations, nor make large approximations as in typical clustering methods. We can also *guarantee* that (to the extent the error model is obeyed) we will find all feasible interpretations of the data in terms of the model.

Associated with each transformation equivalence class (and more generally with each transformation) is a set of matches defined by the following function mapping transformations to match sets. Define

$$\varphi(T) = \{(\mathbf{m}_i, \tilde{\mathbf{d}}_j) | T[\mathbf{m}_i] \in U^j\}$$

to be the set of feature matches aligned by the transformation T. It is the function $\varphi(T)$ that partitions the transformation parameter space forming equivalence classes of transformations. We call the match set $\varphi(T)$ a maximal geometrically-consistent match set. A match set $\mathcal{M}$ is a maximal geometrically-consistent match set (or, a maximal match set) if it is the largest geometrically

consistent match set at some transformation T. This is an important notion because the maximal match sets are the only sets of feature correspondences we need to consider—supersets are not geometrically consistent and subsets correspond to a coarser partition of transformation space and do not accurately represent the geometric consistency (defined below) of a transformation—and there are only a polynomial number of maximal match sets; they are in one-to-one correspondence with the transformation equivalence classes.

Finally, we define the notion of a geometric consistency measure, $\Phi(T)$, characterizing the number of matches feasible for a given transformation T. The simplest measure is a count of the number of matches feasible at a transformation $\Phi^{\mathcal{M}}(T) = |\varphi(T)|$; we use a more sophisticated measure which is the minimum of the number of distinct model features and distinct data features appearing in the feasible match set[26] given by

$$\Phi_T(T) \quad = \quad \texttt{min}\left(\left|\{\mathbf{m}_i; (\mathbf{m}_i, \tilde{\mathbf{d}}_k) \in \varphi(T)\}\right|, \left|\{\mathbf{d}_j; (\mathbf{m}_l, \tilde{\mathbf{d}}_j) \in \varphi(T)\}\right|\right) \tag{1}$$

which approximates the size of a one-to-one feature correspondence between the model and a data subset. A closely related measure of geometric consistency we will define as the bidirectional Hausdorff Fraction[24] given by

$$\Phi_T^H(T) = \texttt{min}\left(\frac{\left|\{\mathbf{m}_i; (\mathbf{m}_i, \tilde{\mathbf{d}}_k) \in \varphi(T)\}\right|}{M}, \frac{\left|\{\mathbf{d}_j; (\mathbf{m}_l, \tilde{\mathbf{d}}_j) \in \varphi(T)\}\right|}{N}\right). \tag{2}$$

Note that these measures are all implicit functions of the uncertainty parameters ϵ_l.

2.2 A Concrete Example: The Case of 3D Models and 2D Data

This paper will focus on using linear bounds on geometric uncertainty. This formulation has been applied to 2D models and transformations and 2D data[5, 7, 12] and 2D models and data and 3D transformations[13, 23]. Here we consider 3D models and transformations and 2D data. The 3D rigid transformation and scaled orthographic projection mapping a model point $\mathbf{m}_i' \in \mathbb{R}^3$ onto a data point $\mathbf{d}_j'$ can be expressed by the affine transformation

$$\mathbf{S}\mathbf{m}_i' + \mathbf{t} = \mathbf{d}_j' \tag{3}$$

where

$$\mathbf{S} = \sigma\mathbf{P}\mathbf{R} = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \end{bmatrix}$$

with $0 < \sigma \in \mathbb{R}$ a positive scale factor, $\mathbf{P}$ a 2×3 projection matrix simply eliminating the third row of $\mathbf{R}$, where $\mathbf{R} \in SO_3$, and $\mathbf{t}, \mathbf{d}_j' \in \mathbb{R}^2$. From $\mathbf{S} = \sigma\mathbf{P}\mathbf{R}$ and the fact the $\mathbf{R}$ is an orthonormal matrix we have the quadratic constraint that

$$\mathbf{S}\mathbf{S}^t = \sigma^2(\mathbf{P}\mathbf{R})(\mathbf{R}^t\mathbf{P}^t) = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}. \tag{4}$$

For convenience we will refer to $\mathbf{S}$ as the linear component of the transformation, and $\mathbf{t}$ the translation component.

In essence the approach is to define a model for geometric uncertainty in the data features which in turn imposes constraints on the geometrically feasible transformations aligning model and data feature sets. This frames geometric feature matching as a problem in computational geometry, specifically constructing and exploring arrangements of constraints on geometrically feasible (i.e. structurally consistent) matchings of model features to data features.

For each data feature formulate a convex polygonal uncertainty region bounding the measurement uncertainty and characterized by a set of linear inequalities[5]. For example, let $\tilde{\mathbf{d}}'_j$ be the *measured* data feature. In the simplest case the uncertainty regions are identical k-sided convex polygons, where the uncertainty region for $\tilde{\mathbf{d}}'_j$ is the set of points $\mathbf{x}'$ satisfying

$$(\mathbf{x}' - \tilde{\mathbf{d}}'_j)^t \hat{\mathbf{n}}_l \quad \geq \quad -\epsilon^l \quad \text{for} \quad l = 1, ..., k$$

where $\hat{\mathbf{n}}_l$ is the unit normal vector and ϵ_l the scalar distance describing each linear constraint for $l = 1, ..., k$. For a transformation on a model feature $\mathbf{m}'_i$ to be feasible with respect to the data feature $\tilde{\mathbf{d}}'_j$ the transformed model feature must satisfy these inequalities and so by substitution the set of feasible transformations for a feature match $\left(\mathbf{m}'_i, \tilde{\mathbf{d}}'_j\right)$ is constrained by the inequalities:

$$(\mathbf{S}\mathbf{m}'_i + \mathbf{t} - \tilde{\mathbf{d}}'_j)^t \hat{\mathbf{n}}_l \geq -\epsilon^l \tag{5}$$

for $l = 1, ..., k$. This describes a polytope of feasible transformation parameters, $\mathcal{F}^{ij}$, mapping $\mathbf{m}'_i$ to within the uncertainty bounds of $\tilde{\mathbf{d}}_j$, where

$$\mathcal{F}^{ij} = \{(\mathbf{S}, \mathbf{t}) | (\mathbf{S}\mathbf{m}'_i + \mathbf{t} - \tilde{\mathbf{d}}'_j)^t \hat{\mathbf{n}}_l \geq -\epsilon^l\}$$

for $l = 1, ..., k$. Each possible feature match $(\mathbf{m}'_i, \tilde{\mathbf{d}}'_j)$ yields such a constraint. Feature matching can be accomplished by searching an arrangement of the boundaries of the polytopes $\mathcal{F}^{ij}$ for all i and j to find sets of transformation parameters with a high geometric consistency measure, i.e. satisfying many of the match constraints. Conceptually the constraints forming the boundaries of the polytopes $\mathcal{F}^{ij}$ partition the parameter space into cells. Exploring the arrangement then involves finding the cells with high geometric consistency. Figure (1) shows an example of this for the case of 2D models and data where the uncertainty regions around the data features are axial squares, and where the allowed transformations are simply translations parameterized by t_1 and t_2. In this case the feasible match polytopes are also squares in the translation space represented by the t_1-t_2 plane.

In the case of 3D affine transformation and orthographic projection (where we relax the quadratic constraints) there are 8 transformation parameters. If there are M model features and N data features then the arrangement of the bounding constraints for each feasible polytopes $\mathcal{F}^{ij}$ has size $O(M^8 N^8)$[19]; constructing and exploring it seems a formidable task. Nonetheless, an important point is that, theoretically, finding *all* geometrically feasible matchings of model and data features within the uncertainty bounds is a polynomial-time problem, in contrast to worst-case exponential-time approaches such as popular feature correspondence-space searches.

3 Towards Practical Application; Developing Mathematical Tools

The previous section introduced the basic formulation for geometric feature matching which facilitates finding all geometrically consistent match sets. Unfortunately constructing and exploring

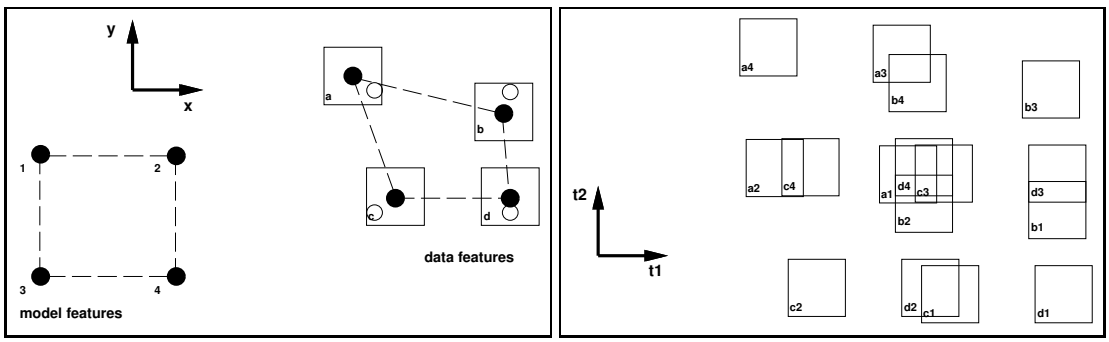

Figure 1: The left panel shows the model features and the data features in the x-y plane. The data features are a translated version of the model features, with some geometric error added. The uncertainty regions for the data features are squares. The open circles near the model features show a feasible translation of the data features. The right panel shows the sets of feasible translations for each feature match. They divide the space up into cells. Each feasible region is labeled according to the features matched. In the center is a place where the four feasible regions corresponding to the correct matches overlap indicating a translation aligning the model and the data to within the uncertainty region.

arrangements of hyperplanes in high dimensional spaces is extremely difficult, and computationally infeasible in this case. There is, however, considerable structure to the feature matching problem which we can exploit to make the computations more feasible and to allow the introduction of careful approximations to greatly reduce the computation required to find large geometrically consistent match sets.

One approach to this is to work directly with the constraints in the s-parameter space spanned by the coordinates s_{ij}. This approach was outlined for the 3D case previously[13] and is detailed in section 8. This section introduces an elegant formulation which through convenient choice of representations and careful approximation reduces this from a constraint problem in an 8D parameter space to a constraint problem in a 2D parameters space which is much more computationally feasible.

3.1 The Image of 3D Points Under Scaled-Orthography

Let $\{\mathbf{m}'_i\}_{i=0}^{M-1}$ be a set of 3D model points, and $\{\mathbf{d}'_i\}_{i=0}^{M-1}$ be the set of points forming their image without error from an arbitrary viewpoint under scaled-orthography such that $\mathbf{S}\mathbf{m}'_i + \mathbf{t} = \mathbf{d}'_i$. Distinguish three model points $\mathbf{m}'_{\pi_0}$, $\mathbf{m}'_{\pi_1}$, and $\mathbf{m}'_{\pi_2}$ and their corresponding image points $\mathbf{d}'_{\pi_0}$, $\mathbf{d}'_{\pi_1}$, and $\mathbf{d}'_{\pi_2}$ which we will call primary model features and primary data features, respectively. Define $\mathbf{m}'_i \neq \mathbf{m}'_{\pi_0}, \mathbf{m}'_{\pi_1}, \mathbf{m}'_{\pi_2}$ and $\mathbf{d}'_i \neq \mathbf{d}'_{\pi_0}, \mathbf{d}'_{\pi_1}, \mathbf{d}'_{\pi_2}$ secondary model features and secondary data features, respectively. Finally, define $\mathbf{m}_i = (\mathbf{m}'_i - \mathbf{m}'_{\pi_0})$ and $\mathbf{d}_i = (\mathbf{d}'_i - \mathbf{d}'_{\pi_0})$ for all i which defines new coordinate frames for the model and the image data with $\mathbf{m}'_{\pi_0}$ and $\mathbf{d}'_{\pi_0}$ as origins, respectively. The new primary features are thus $\mathbf{m}_{\pi_1}$, $\mathbf{m}_{\pi_2}$, $\mathbf{d}_{\pi_1}$, and $\mathbf{d}_{\pi_2}$ and we now have $\mathbf{S}\mathbf{m}_i = \mathbf{d}_i$ for all i.

Express the model point $\mathbf{m}_i$ using $\mathbf{m}_{\pi_1}$, $\mathbf{m}_{\pi_2}$, and $\hat{\mathbf{n}} = \frac{\mathbf{m}_{\pi_1} \times \mathbf{m}_{\pi_2}}{|\mathbf{m}_{\pi_1} \times \mathbf{m}_{\pi_2}|}$ as basis vectors[4]. Assuming $\mathbf{m}_{\pi_1}$ and $\mathbf{m}_{\pi_2}$ are linearly independent there exist three constants a_i, b_i, and c_i such that

$$\mathbf{m}_i = a_i \mathbf{m}_{\pi_1} + b_i \mathbf{m}_{\pi_2} + c_i \hat{\mathbf{n}} = \begin{bmatrix} \mathbf{m}_{\pi_1} & \mathbf{m}_{\pi_2} \end{bmatrix} \begin{bmatrix} a_i \\ b_i \end{bmatrix} + c_i \hat{\mathbf{n}}.$$

The original orientation of the model features is arbitrary, so without loss of generality let the model be oriented such that

$$[\,\mathbf{m}_{\pi_1} \quad \mathbf{m}_{\pi_2}\,] = \begin{bmatrix} m^x_{\pi_1} & m^x_{\pi_2} \\ m^y_{\pi_1} & m^y_{\pi_2} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{m}}_{\pi_1} & \bar{\mathbf{m}}_{\pi_2} \\ 0 & 0 \end{bmatrix}$$

where $\bar{\mathbf{m}}_{\pi_1}, \bar{\mathbf{m}}_{\pi_2} \in \mathbb{R}^2$. We will call this a canonical model orientation with respect to the particular primary model triple.[2] We can think of $\mathbf{m}_i$ as having a planar component, $[\,\mathbf{m}_{\pi_1} \quad \mathbf{m}_{\pi_2}\,][\,a_i \quad b_i\,]^t$, lying in the plane of the three primary model features $\mathbf{m}'_{\pi_0}, \mathbf{m}'_{\pi_1}, \mathbf{m}'_{\pi_2}$ and a non-planar component, $c_i\hat{\mathbf{n}}$, perpendicular to this plane. In the canonical case we simply have

$$\mathbf{m}_i = [\,\mathbf{m}_{\pi_1} \quad \mathbf{m}_{\pi_2}\,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} + c_i\hat{\mathbf{z}}. \tag{6}$$

The imaging transformation for the primary feature matches can be expressed simply as

$$\mathbf{S}\,[\,\mathbf{m}_{\pi_1} \quad \mathbf{m}_{\pi_2}\,] = \mathbf{S}_4\,[\,\bar{\mathbf{m}}_{\pi_1} \quad \bar{\mathbf{m}}_{\pi_2}\,] = [\,\mathbf{d}_{\pi_1} \quad \mathbf{d}_{\pi_2}\,]$$

with

$$\mathbf{S}_4 = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} \mathbf{S}_4 & \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \end{bmatrix} \tag{7}$$

where we have distinguished the third column of $\mathbf{S}$ by defining

$$\vec{\xi} = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \begin{bmatrix} s_{13} \\ s_{23} \end{bmatrix}$$

because ξ_1 and ξ_2 are fundamental parameters in this formulation, independent of the orientation of the primary model features[15]. These expressions then lead to a very simple expression for the image of any secondary model feature $\mathbf{m}_i$ under the imaging transformation $\mathbf{S}$ which aligns the primary features.

$$\mathbf{S}\mathbf{m}_i = c_i \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + [\,\mathbf{d}_{\pi_1} \quad \mathbf{d}_{\pi_2}\,] \begin{bmatrix} a_i \\ b_i \end{bmatrix}. \tag{8}$$

The image of $\mathbf{m}_i$ under the matrix $\mathbf{S}$ thus has two components. The component due to the planar component of $\mathbf{m}_i$, given by $[\,\mathbf{d}_{\pi_1} \quad \mathbf{d}_{\pi_2}\,][\,a_i \quad b_i\,]^t$, and the component due to the non-planar component of $\mathbf{m}_i$, given by $c_i\,[\,\xi_1 \quad \xi_2\,]^t$. The former can be seen as simply the planar component of $\mathbf{m}_i$ expressed in the affine basis $[\,\mathbf{d}_{\pi_1} \quad \mathbf{d}_{\pi_2}\,]$. The parameters s_{11}, s_{12}, s_{21}, and s_{22} operate on the planar component of $\mathbf{m}_i$ and ξ_1 and ξ_2 operate on the non-planar component.

We will call equation (8) the reprojection equation characterizing the reprojection of the model point $\mathbf{m}_i$ into the image under the transformation mapping $\mathbf{m}_{\pi_1}$ to $\mathbf{d}_{\pi_1}$ and $\mathbf{m}_{\pi_2}$ to $\mathbf{d}_{\pi_2}$. Under rigid 3D motion and scaled-orthographic projection, the correspondence between the primary model features, $\mathbf{m}'_{\pi_0}$, $\mathbf{m}'_{\pi_1}$, and $\mathbf{m}'_{\pi_2}$ and the primary data features $\mathbf{d}'_{\pi_0}$, $\mathbf{d}'_{\pi_1}$, and $\mathbf{d}'_{\pi_2}$ is sufficient to compute an almost unique 3D rigid transformation, $\mathbf{S}$ and $\mathbf{t}$, on the model features which has been called the

[2] A rotation matrix $\mathbf{R}_c$ producing such a canonical model orientation can easily be constructed as $\mathbf{R}_c = \begin{bmatrix} \frac{\mathbf{m}'_{\pi_1}}{|\mathbf{m}'_{\pi_1}|} & \frac{\hat{\mathbf{n}}\times\mathbf{m}'_{\pi_1}}{|\mathbf{m}'_{\pi_1}|} & \hat{\mathbf{n}} \end{bmatrix}^t$

alignment transformation. The computation of the alignment transformation has been considered in detail before[25, 4]; the formulation presented here is considered in detail by Cass[15] where its relationship to affine model indexing and affine structure from motion is explored. Note that by choosing $\mathbf{m}'_{\pi_0}$ and $\mathbf{d}'_{\pi_0}$ as origins we have eliminated the translation component of the transformation, for now.

When the transformation given by $\mathbf{S}$ and $\mathbf{t}$ corresponds to 3D affine transformation and orthographic projection, the components $[\,\xi_1 \quad \xi_2\,]^t$ can take on any value corresponding to different viewpoints and affine transformations. When the correspondence between the primary features is used to compute a *rigid* alignment transformation then the components $[\,\xi_1 \quad \xi_2\,]^t$ must satisfy two quadratic constraints to produce rigid 3D motion and scaled orthographic projection. This follows directly[15] from equations (4) and (8) and is considered in an appendix in section 10.

3.2 Incorporating the Effects Of Uncertainty

Central to the reprojection equation and the computation of the rigid three-point alignment transformation are the three primary model and data feature matches. The effect of geometric uncertainty in these primary features is significant and must be handled carefully, in addition to uncertainty in the secondary model features. We will follow an approach representing the measured data feature and the error as explicit terms[21]. When there is geometric uncertainty in the measured position of the data features the transformation mapping model feature $\mathbf{m}_i$ to the measured data feature $\tilde{\mathbf{d}}_j$ must satisfy

$$\mathbf{S}\mathbf{m}'_i + \mathbf{t} = \tilde{\mathbf{d}}'_j = \mathbf{d}'_j + \mathbf{e}'_j,$$

where again $\mathbf{d}_j$ represents the (unknown and unmeasurable) true position of the data feature, $\mathbf{e}'_j$ the measurement error, and $\mathbf{d}'_j + \mathbf{e}'_j = \tilde{\mathbf{d}}'_j$ represents the *observed* location of the data feature[21]. As before we match $\mathbf{m}'_{\pi_0}$ and $\mathbf{d}'_{\pi_0}$ as corresponding origins in the model and data coordinate frames, respectively, and define $\mathbf{m}_i = (\mathbf{m}'_i - \mathbf{m}'_{\pi_0})$ for all i and $\tilde{\mathbf{d}}_j = (\tilde{\mathbf{d}}'_j - \tilde{\mathbf{d}}'_{\pi_0})$ for all j. We then have

$$\mathbf{S}\mathbf{m}_i = \tilde{\mathbf{d}}_j = \mathbf{d}_j + \mathbf{e}_j$$

where we define $\mathbf{e}_j = \mathbf{e}'_j - \mathbf{e}'_{\pi_0}$.

When the reprojection or alignment transformation is computed from the *measured* geometry of the primary data features $\tilde{\mathbf{d}}_{\pi_1}$ and $\tilde{\mathbf{d}}_{\pi_2}$ the reprojection of a secondary model feature is given by

$$\mathbf{S}\mathbf{m}_i = c_i \begin{bmatrix} \tilde{\xi}_1 \\ \tilde{\xi}_2 \end{bmatrix} + [\,\tilde{\mathbf{d}}_{\pi_1} \quad \tilde{\mathbf{d}}_{\pi_2}\,] \begin{bmatrix} a_i \\ b_i \end{bmatrix}.$$

The effect of the error on the reprojection of a model point $\mathbf{m}_i$ can be decomposed in terms of the effect on the planar component $[\,\tilde{\mathbf{d}}_{\pi_1} \quad \tilde{\mathbf{d}}_{\pi_2}\,] \begin{bmatrix} a_i \\ b_i \end{bmatrix}$ and the effect on the non-planar component $c_i \begin{bmatrix} \tilde{\xi}_1 \\ \tilde{\xi}_2 \end{bmatrix}$, where we have defined $\tilde{\xi}_1 = \xi_1(\tilde{\mathbf{d}}_{\pi_1}, \tilde{\mathbf{d}}_{\pi_2})$ and $\tilde{\xi}_2 = \xi_2(\tilde{\mathbf{d}}_{\pi_1}, \tilde{\mathbf{d}}_{\pi_2})$ as the rigid transformation components computed from the measured primary data features (see Appendix 10.1). The planar component of the reprojection has a simple linear form[21] given by

$$[\,\tilde{\mathbf{d}}_{\pi_1} \quad \tilde{\mathbf{d}}_{\pi_2}\,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} = [\,\mathbf{d}_{\pi_1} \quad \mathbf{d}_{\pi_2}\,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} + [\,\mathbf{e}_{\pi_1} \quad \mathbf{e}_{\pi_2}\,] \begin{bmatrix} a_i \\ b_i \end{bmatrix}. \tag{9}$$

The non-planar component of the reprojection has a complicated non-linear dependence on the uncertainty in the primary data features[15] which is a problem when computing $\tilde{\xi}_1$ and $\tilde{\xi}_2$ for a rigid transformation. We will only be concerned with 3D affine transformation, and will constrain ξ_1 and ξ_2 without explicitly computing them.

4 Geometric Feature Matching: Applying the Tools

We now consider a sequence of formulations which make successively more approximations to reduce the dimensionality of the geometric constraints. We emphasize that these approximations are *conservative* in that they will never exclude a feasible transformation. They are also careful approximations in that they do not include *too many* infeasible transformations. These formulations illustrate that the geometric feature matching problem has both a purely combinatorial component as well as a geometric component and the different balance between these components.

4.1 An 8D Constraint Problem

We will rederive the transformations constraints of section 2.2 by starting with the equation for an aligned feature match $\mathbf{S}\mathbf{m}'_i + \mathbf{t} - \mathbf{d}'_j = 0$ or equivalently incorporating error $\mathbf{S}\mathbf{m}'_i + \mathbf{t} - \tilde{\mathbf{d}}'_j = -\mathbf{e}'_j$. From the definition of the uncertainty bounds we have $\hat{\mathbf{n}}_l^t \mathbf{e}'_p \leq \epsilon^l$, for $l = 1, ..., k$ and for all p, which directly leads to the inequalities (5). These inequalities describe an 8D constraint arrangement of size $O(M^8 N^8)$ which can be explored in $O(M^8 N^8)$ time[19].

4.2 A 6D Constraint Problem

Next hypothesize the correspondence between one model feature $\mathbf{m}'_{\pi_0}$ and one data feature $\tilde{\mathbf{d}}'_{\rho_0}$ and let these points be the origin for the model and image coordinate frames, respectively. We subscript the primary data feature $\tilde{\mathbf{d}}_{\rho_0}$ differently than the model feature $\mathbf{m}_{\pi_0}$ to emphasize that their correspondence to one another is hypothesized, not known. As before we define $\mathbf{m}_i = (\mathbf{m}'_i - \mathbf{m}'_{\pi_0})$ and $\tilde{\mathbf{d}}_i = (\tilde{\mathbf{d}}'_i - \tilde{\mathbf{d}}'_{\rho_0})$ for all i, and the transformation aligning $\mathbf{m}_i$ and $\tilde{\mathbf{d}}_j$ must satisfy $\mathbf{S}\mathbf{m}_i - \mathbf{d}_j = 0$ or equivalently $\mathbf{S}\mathbf{m}_i - \tilde{\mathbf{d}}_j = -\mathbf{e}_j$ where we define $\mathbf{e}_j = \mathbf{e}'_j - \mathbf{e}'_{\rho_0}$. Using again the uncertainty bound $\hat{\mathbf{n}}_l^t \mathbf{e}'_p \leq \epsilon^l$, for $l = 1, ..., k$ and for all p we then have constraints for a single match on the 6 s-parameters which comprise the linear component of the transformation: $(\mathbf{S}\mathbf{m}_i - \tilde{\mathbf{d}}_j)^t \hat{\mathbf{n}}_l \geq -2\epsilon^l$ for $l = 1, ...$ This is the formulation used in previous work[13]. This constraint arrangement is of size $O(M^6 N^6)$, and we see that we have shifted around where the computation must be done. For each choice of π_0 and ρ_0 we analyze the arrangement of s-parameter constraints, of size $O(M^6 N^6)$, but must perform a combinatorial search through the MN possible different primary matches $(\mathbf{m}_{\pi_0}, \tilde{\mathbf{d}}_{\rho_0})$ for $\pi_0 = 1, ..., M-1$, $\rho_0 = 1, ..., N-1$, and for each match do the s-parameter constraint analysis described above. This is also an approximation because each uncertainty region shares the error vector $\mathbf{e}_{\rho_0}$ which can only take on one global value. Thus we are computing approximate geometric consistency, but conservatively in the sense that the bounds are conservative thus we will never underestimate true global consistency in the arrangement.

4.3 A 2D Constraint Problem

Next assume the correspondence between three primary model and data features, $(\mathbf{m}'_{\pi_0}, \tilde{\mathbf{d}}_{\rho_0})$, $(\mathbf{m}'_{\pi_1}, \tilde{\mathbf{d}}_{\rho_1})$, and $(\mathbf{m}'_{\pi_2}, \tilde{\mathbf{d}}_{\rho_2})$, and represent the model features in terms of the natural affine coordinate frame as discussed in section 3.1. In order for a feature match $(\mathbf{m}_i, \mathbf{d}_j)$ to be feasible we must have that

$$c_i \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + [\, \mathbf{d}_{\pi_1} \quad \mathbf{d}_{\pi_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} = \mathbf{d}_j$$

for some value of the viewpoint parameters ξ_1 and ξ_2. Using the actual observed data features and accounting for uncertainty we must have

$$c_i \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + [\, \tilde{\mathbf{d}}_{\pi_1} \quad \tilde{\mathbf{d}}_{\pi_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} - \tilde{\mathbf{d}}_j = [\, \mathbf{e}_{\pi_1} \quad \mathbf{e}_{\pi_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} - \mathbf{e}_j \tag{10}$$

and expanding the individual error terms[21] we get

$$c_i \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + [\, \tilde{\mathbf{d}}_{\pi_1} \quad \tilde{\mathbf{d}}_{\pi_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} - \tilde{\mathbf{d}}_j = -\mathbf{e}'_j + \mathbf{e}'_{\pi_0} + [\, \mathbf{e}'_{\pi_1} \quad \mathbf{e}_{\pi_2}{}' \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} - [\, \mathbf{e}'_{\pi_0} \quad \mathbf{e}_{\pi_0}{}' \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix}. \tag{11}$$

A bound on the uncertainty region for the sum of the errors on the right hand side of equation (11) is given by the Minkowsky sum of the individual uncertainty regions for $\mathbf{e}'_{\rho_0}$, $\mathbf{e}'_{\rho_1}$, $\mathbf{e}'_{\rho_2}$, and $\mathbf{e}'_j$, which leads to constraints for each match on the two viewpoint parameters, ξ_1 and ξ_2.[3]

$$(c_i \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + [\, \tilde{\mathbf{d}}_{\rho_1} \quad \tilde{\mathbf{d}}_{\rho_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} - \tilde{\mathbf{d}}_j)^t \hat{\mathbf{n}}_l \;\; \geq \;\; -\epsilon^l (|a_i + b_i - 1| + |a_i| + |b_i| + 1) \tag{12}$$

Inequality (12) represents conservative constraints on the parameters ξ_1 and ξ_2 feasible for feature match $(\mathbf{m}_i, \tilde{\mathbf{d}}_j)$ when the primary model and data features are approximately aligned; these constraints are based directly on bounds on measurement error for $\mathbf{d}'_{\rho_0}$, $\mathbf{d}'_{\rho_1}$, $\mathbf{d}'_{\rho_2}$, and $\mathbf{d}'_j$.

After this change to affine bases we simply analyze a 2D constraint arrangement for the two components ξ_1 and ξ_2 which is now an arrangement of convex constraint polygons, one for each secondary feature match. Each polygon's shape is a scaled copy of the basic data uncertainty region and its size is roughly proportional to $(|a_i| + |b_i| + 1)/c_i$. This arrangement of MN such polygons is now only of size $O(M^2N^2)$, however we increased the combinatorial component to time $O(M^3N^3)$ because we must search over the possible triples of primary model and data features which form the affine bases. These primary matches must be correctly matched to find geometrically consistent secondary matches. This formulation for deriving constraints on ξ_1 and ξ_2 for each match is closely related to an approach working directly in the space of s-parameters s_{ij} using the constraints of section 4.2 and making a careful projection of the constraints onto a lower dimensional space[14]. This is described in section 8.

In section 5 we will use these 2D constraints for a very simple method for feature matching. First we will note some special cases of this formulation.

[3] The bound on the feasible viewpoint parameters ξ_1 and ξ_2 is essentially the same as the bounds on the reprojection of a secondary model feature by an alignment transformation computed from three primary matches in the case of *coplanar* model features[21].

4.4 The Case of 2D Affine Transformations: Planar Objects in 3D

In the special case where the model features are coplanar, we have that $c_i = 0$ for all i. In this case the transformation corresponds to a 2D affine transformation of planar point sets. In this case we can use the vector $\mathbf{m}_{\pi_1}$ and its perpendicular $\mathbf{m}_{\pi_1}^{\perp}$ as a basis to represent the co-planar model features where

$$\mathbf{m}_i = [\, \mathbf{m}_{\pi_1} \quad \mathbf{m}_{\pi_1}^{\perp} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix}. \tag{13}$$

Expressing the fact that $\mathbf{m}_{\pi_1}$ is mapped to $\mathbf{d}_{\rho_1}$ and the virtual point $\mathbf{m}_{\pi_1}^{\perp}$ is mapped *anywhere*, let's say $\mathbf{m}_{\pi_1}^{\perp}$ is mapped to $\vec{\eta}$, so

$$\mathbf{S}\,[\, \mathbf{m}_{\pi_1} \quad \mathbf{m}_{\pi_1}^{\perp} \,] = [\, \mathbf{d}_{\rho_1} \quad \vec{\eta} \,]$$

and the image of the model point is characterized by two parameters $\vec{\eta} = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}$ and given by

$$\mathbf{S}\mathbf{m}_i = [\, \mathbf{d}_{\rho_1} \quad \vec{\eta} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix}.$$

Here

$$\mathbf{S} = \mathbf{S}_4 = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix}.$$

Consider the equation matching $\mathbf{m}_i$ to $\mathbf{d}_j$

$$[\, \mathbf{d}_{\rho_1} \quad \vec{\eta} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} = \mathbf{d}_j$$

or equivalently

$$[\, \tilde{\mathbf{d}}_{\rho_1} \quad \vec{\eta} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} = \tilde{\mathbf{d}}_j - \mathbf{e}_j + a_i \mathbf{e}_{\rho_1}$$

which as before leads to the constraints

$$([\, \tilde{\mathbf{d}}_{\rho_1} \quad \vec{\eta} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} - \tilde{\mathbf{d}}_j)^t \hat{\mathbf{n}}_l \quad \geq \quad -\epsilon^l (|a_i - 1| + |a_i| + 1) \tag{14}$$

We now have a combinatorial component of the algorithm where we search through primary feature match *pairs* $(\mathbf{m}_{\pi_0}, \tilde{\mathbf{d}}_{\rho_0})$ and $(\mathbf{m}_{\pi_1}, \tilde{\mathbf{d}}_{\rho_1})$. We do a combinatorial search through pairs of primary features, and for each perform an anlysis of a 2D arrangement of constraint polygons, one for each secondary feature match, in the η_1-η_2 plane, each polygon's shape is a scaled copy of the basic data uncertainty region and its size is roughly proportional to $(|a_i| + 1)/b_i$. This is closely related to the approach described previously by the author[13, 14]. A related approach is described by Hopcroft[23].

4.4.1 The Special Case of Planar Objects in 2D

In the 2D case of scaled, rigid planar motion the model features are transformed by

$$\mathbf{S} = \mathbf{S}_2 = \begin{bmatrix} s_1 & -s_2 \\ s_2 & s_1 \end{bmatrix}$$

with a little manipulation we see

$$\mathbf{S}\mathbf{m}_i = \begin{bmatrix} m_i^x & -m_i^y \\ m_i^y & m_i^x \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = [\, \mathbf{m}_i \quad \mathbf{m}_i^\perp \,] \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \mathbf{d}_j = \tilde{\mathbf{d}}_j - \mathbf{e}_j.$$

From this we see that the parameters s_1 and s_2 for match $(\mathbf{m}_i, \tilde{\mathbf{d}}_j)$ are linearly related to the error $\mathbf{e}_j$. Bounding $\mathbf{e}_j$ leads directly to halfplane constraints on s_1 and s_2 which in the case of axial square bounds on $\mathbf{e}_j$ take the form of square bounds in the s_1-s_2 for each match $(\mathbf{m}_i, \tilde{\mathbf{d}}_j)$, where the square's orientation is related to $\mathbf{m}_i$ and it's size related to $|\mathbf{m}_i|$. We analyze this arrangement of squares directly to find feasible values of the transformation parameters s_1 and s_2[12].

5 An Algorithm for Geometric Matching

We can now pull together the ideas from the previous sections to develop an algorithm for geometric matching of 3D models to 2D data. A basic algorithm for geometric feature matching consists of the following steps. Perform a combinatorial search through triples of feature matches each defining affine bases for the model and data points. For each such primary match triple, compute for each secondary feature match the constraint polygon for the set of feasible transformation parameters ξ_1 and ξ_2 and analyze the 2D constraint arrangement to find match sets with a high conservative estimate of geometric consistency. When large approximately consistent match sets are found we can optionally perform a more careful search of the feasible transformations using more strict bounds; this will be explained in section 5.3. Deriving the constrains on ξ_1 and ξ_2 can be viewed as a conservative approximate method of finding regions of potentially high geometric consistency for the full constraint arrangement characterized by inequalities (5). This is done by careful approximation and projection to lower dimensional subspaces.

5.1 Identical Axial Square Uncertainty Regions

Consider the above constraint formulation using axial square uncertainty bounds for the position of the data features characterized by $|\hat{\mathbf{x}}^t \mathbf{e}'_j| \leq \epsilon$ and $|\hat{\mathbf{y}}^t \mathbf{e}'_j| \leq \epsilon$ for all j. Assume we have selected a triple of primary feature matches yielding $(\mathbf{m}_{\pi_1}, \tilde{\mathbf{d}}_{\rho_1})$ and $(\mathbf{m}_{\pi_2}, \tilde{\mathbf{d}}_{\rho_2})$ as primary matches. In this case equation (12) yields

$$|(c_i \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + [\, \tilde{\mathbf{d}}_{\rho_1} \quad \tilde{\mathbf{d}}_{\rho_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} - \tilde{\mathbf{d}}_j)^t \hat{\mathbf{x}}| \leq \epsilon(|a_i + b_i - 1| + |a_i| + |b_i| + 1) \tag{15}$$

$$|(c_i \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + [\, \tilde{\mathbf{d}}_{\rho_1} \quad \tilde{\mathbf{d}}_{\rho_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} - \tilde{\mathbf{d}}_j)^t \hat{\mathbf{y}}| \leq \epsilon(|a_i + b_i - 1| + |a_i| + |b_i| + 1) \tag{16}$$

as constaints on the *unknown* vector $\begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}$ feasible with respect to the secondary feature match $(\mathbf{m}_i, \tilde{\mathbf{d}}_j)$. The set of feasible viewpoint parameters $\begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}$ for the feature match $(\mathbf{m}_i, \tilde{\mathbf{d}}_j)$ is bounded by a square of side $\frac{2\epsilon}{c_i}(|a_i + b_i - 1| + |a_i| + |b_i| + 1)$ centered at the point $\frac{1}{c_i}(\tilde{\mathbf{d}}_j - [\, \tilde{\mathbf{d}}_{\rho_1} \quad \tilde{\mathbf{d}}_{\rho_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix})$ in the ξ_1-ξ_2 plane.

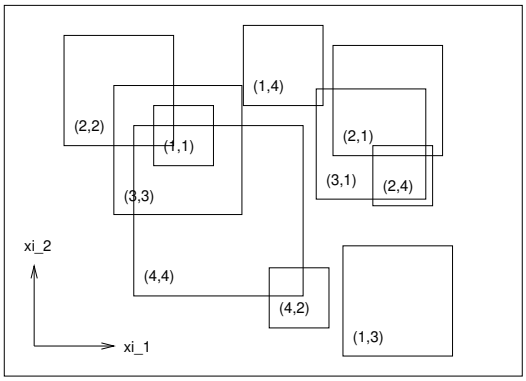

Figure 2: xi-plane A set of constraint squares in the ξ_1-ξ_2 plane, one for each model data feature match (denoted (i, j) in the figure). Regions of the plane where several squares overlap indicate a range of parameters for which many matches are approximately aligned.

For every secondary match $(\mathbf{m}_i, \tilde{\mathbf{d}}_j)$ we have such a square constraint on the feasible non-linear component $\begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}$. There are $(M-3)(N-3)$ of them forming an arrangement with $O(M^2N^2)$ cells in the ξ_1-ξ_2 plane. See figure (2). Each cell of this arrangement corresponds to a set of values for the parameters $[\, \xi_1 \quad \xi_2 \,]^t$ for which a set of feature matches are consistent with respect to these approximate bounds. Finding cells consistent for many feature matches yields match sets and transformations which are very good candidates for large, fully geometrically consistent match sets. An important point, however, is that although there are $O(M^2N^2)$ cells finding the cells of maximal coverage in this arrangment can be done in $O(MN \lg MN)$ time as described in Section 5.2, depending on our measure of consistency. Three which can be computed in $O(MN \lg MN)$ time are: the widely used measures consiting of the count of number of matches consistent with these constraints at some point; the largest number of distinct model features aligned with a data feature; and the largest number of distinct data features aligned with a model feature. It is more difficult to compute the measure $\Phi(T)$, with $T = [\, \xi_1 \quad \xi_2 \,]^t$, described in equation (1). We do not know of an $O(MN \lg MN)$ time algorithm for this, although we present a heuristic algorithm with very good performance.

Note that the approximations involved in projecting to the ξ_1-ξ_2 plane are conservative in the sense they will never miss a large geometrically consistent match set of model and data features. Thus if there are no cells in this arrangement have high consistency, then no further exploration need be taken with the particular primary matches.

For the case of planar objects the constraints are squares of side $\frac{2\epsilon}{b_i}(|a_i - 1| + |a_i| + 1)$ centered at the point $\frac{1}{b_i}(\tilde{\mathbf{d}}_j - a_i\tilde{\mathbf{d}}_{\rho_1})$ in the η_1-η_2 plane.

5.1.1 Gaussian Approximation to Uncertainty Bounds

The bounds derived above are extremely conservative, providing an absolute bound for the case of a uniform, square distribution for the uncertainty in the primary data features. The range of possible values for ξ_1 and ξ_2 for a given secondary match is a function of the three uncertainty vectors $\mathbf{e}'_{\rho_0}$, $\mathbf{e}'_{\rho_1}$, and $\mathbf{e}'_{\rho_2}$. If these vectors are uniformly distributed in the square bounds, the distribution of

feasible parameters ξ_1 and ξ_2 will then be a piecewise quadratic function in the plane. The tails of this distribution are much less likely than the middle regions, thus using bounding squares in the ξ-plane is rather inaccurate.

Rather than computing the distribution of $\vec{\xi}$ exactly (which is straighforward because the distributions oare decomposeable into products of 1D distributions) we utilized a simple approximation to get a slightly tighter and more accurate bound on feasible values of ξ_1 and ξ_2 as follows. Approximate the uniform distribution of $\mathbf{e}'_{\rho_0}$, $\mathbf{e}'_{\rho_1}$, and $\mathbf{e}'_{\rho_2}$ by assuming they are identically distributed Gaussian random variables with variance $\sigma_e^2 = \epsilon^2/4$. So our bounded regions enclose approximately distances within 2σ. The vector $\vec{\xi} = \begin{bmatrix} \xi_1 & \xi_2 \end{bmatrix}^t$ is then a linear function of these random variables and is thus also a Gaussian random variable. The variance of $\vec{\xi}$ is then $\sigma_{\xi_i}^2 = \frac{4\epsilon^2}{c_i^2}[(a_i+b_i-1)^2+a_i^2+b_i^2+1)]$. If we choose the bounding constraint square for ξ_1 and ξ_2 to be $2\sigma_{\xi_i}$ we have a tighter, though approximate bound on ξ_1 and ξ_2; the advantage is that the fairly unlikely tails are excluded. In the case of planar objects we can get a similar bound on η_1 and η_2 using $\sigma_{\eta_i}^2 = \frac{4\epsilon^2}{b_i^2}[(a_i-1)^2+a_i^2+1)]$, and taking the constraint squares to be of size $2\sigma_{\eta_i}$. This is just an alternative way to derive the constraint squares on feasible parameters ξ_1 and ξ_2 (or η_1 and η_2); the same geometric algorithms are then utilized.

5.2 Analyzing the Planar Constraint Arrangement

This section describes the algorithm for analyzing the arrangement of *axial squares* in the ξ_1-ξ_2 plane constraining feasible values of ξ_1 and ξ_2. There are mainly two consistency measures we will consider for a match set at a particular transformation, defined in section 2 and equation (1). Define the set $\mathcal{F}_\xi^{ij}$ to be the set of parameters (ξ_1, ξ_2) satisfying equation (12) for feature match $(\mathbf{m}_i, \tilde{\mathbf{d}}_j)$; $\mathcal{F}_\xi^{ij}$ is an axial square in the ξ_1-ξ_2 plane. Also define

$$\varphi_\xi(\xi_1, \xi_2) = \{(\mathbf{m}_i, \tilde{\mathbf{d}}_j) | (\xi_1, \xi_2) \in \mathcal{F}_\xi^{ij}\}.$$

The first consistency measure is the cardinality of the match set consistent with respect to the ξ_1-ξ_2 constraints, that is the count of the number of feature match constraints geometrically consistent for that transformation. This is defined as $\Phi_\xi^{\mathcal{M}} = |\varphi_\xi(\xi_1, \xi_2)|$. The second is a quantity consisting of the minimum of the number of distinct model features appearing in the match set and the number of distinct data features appearing in the match set given by

$$\Phi_\xi(\xi_1, \xi_2) = \mathtt{min}\left(\left|\{\mathbf{m}_i; (\mathbf{m}_i, \tilde{\mathbf{d}}_k) \in \varphi_\xi(\xi_1, \xi_2)\}\right|, \left|\{\tilde{\mathbf{d}}_j; (\mathbf{m}_l, \tilde{\mathbf{d}}_j) \in \varphi_\xi(\xi_1, \xi_2)\}\right|\right) \tag{17}$$

which is closely related to an approximation (because our constraints are approximate) of the bidirectional Hausdorff fraction.

Computing the match count $\Phi_\xi^{\mathcal{M}}$ is straightforward. Compute the MN axial squares for the matches as described in section 5.1. We seek a point in the ξ_1-ξ_2 plane of maximal coverage by the squares. This can be computed, for MN axial rectangles in time $O(MN \lg MN)$ using a segment tree[33].

To compute $\Phi_\xi(\xi_1, \xi_2)$ we use a heuristic algorithm with the same $O(MN \lg MN)$ time behavior. It is heuristic in that it can only be guaranteed to produce a lower bound to the maximal value of $\Phi_\xi(\xi_1, \xi_2)$, but performs very well empirically. To compute it exactly we could use a variant of an segment tree algorithm which would then require $O(M^2N^2 \lg MN)$ time and linear space.

We start by slightly extending a technique of Chew and Kedem[16]. Define

$$\mathcal{F}^{\mathbf{m}_i} = \bigcup_j \mathcal{F}_\xi^{ij}$$

for each model feature $\mathbf{m}_i$. Similarly define

$$\mathcal{F}^{\mathbf{d}_j} = \bigcup_j \mathcal{F}_\xi^{ij}$$

for each data feature $\tilde{\mathbf{d}}_j$. The set $\mathcal{F}^{\mathbf{m}_i}$ is the union of N axial squares. Because these are squares, the number of segments forming the boundary edges of the union is $O(N)$. Similarly the set $\mathcal{F}^{\mathbf{d}_j}$ is the union of M axial squares, is of size $O(M)$ and can be computed in $O(M)$ time. Next partition each object $\mathcal{F}^{\mathbf{m}_i}$ into $O(N)$ non-overlapping axial rectangles in $O(N \lg N)$ time. This can be done using a modification of a segment tree algorithm. Informally, each vertical boundary segment of the union potentially forms a vertical edge to a rectangle in the partition where we construct new horizontal edges as needed to form partition rectangles. Each of the M sets $\mathcal{F}^{\mathbf{m}_i}$ is partitioned into $O(N)$ rectangles, so all together this forms $O(MN)$ axial rectangles, each associated with a particular model feature $\mathbf{m}_i$. The non-overlapping partitioning means that no two overlapping rectangles are associated with the same model feature (with the subtlety that the edges of the adjacent partition rectangles overlap).

Similarly partition each of the N objects $\mathcal{F}^{\mathbf{d}_j}$ into $O(M)$ non-overlapping axial rectangles in $O(M \lg M)$ time. This forms another $O(MN)$ axial rectangles, each associated with a particular image feature $\mathbf{d}_j$. Finally, analyze together the arrangement of these two groups of $O(MN)$ axial rectangles, each labeled with either a model feature $\mathbf{m}_i$ or a data feature $\mathbf{d}_j$, to find the maximal value of $\Phi_\xi(\xi_1, \xi_2)$ again using a segment tree. At each point of the arrangement, Φ_ξ is just the minimum of the number model rectangles covering the point and the number of data rectangles covering the point. The heuristic computation is a straightforward modification of the basic segment tree algorithm which maintains the value best possible value of $\Phi_\xi(\xi_1, \xi_2)$ for each subtree of the segment tree in the natural way in time $O(MN \lg MN)$ time. This extention is detailed in Appendix 10.2.

In summary, for each triple of primary matches we can compute an approximation of the geometric consistency, the maximal value of $\Phi_\xi(\xi_1, \xi_2)$ over the arrangement of constraint squares, in $O(MN \lg MN)$ time. We can easily then find a point in the arrangement achieving this value. For 3D objects the overall algorithm then requires time $O(M^4N^4 \lg MN)$, or $O(MN^4 \lg MN)$ expected time; for planar 3D objects and affine transformations the algorithm then requires time $O(M^3N^3 \lg MN)$, or $O(MN^3 \lg MN)$ expected time (see section 6).

We could stop at this point and take the transformation parameters s_{11}, s_{12}, s_{21}, s_{22} aligning the primary matches together with ξ_1 and ξ_2. This would be similar to a careful alignment approach[25, 4], although bringing to bear more constraint in one sense, and less in another (see section 8). By choosing $\mathbf{m}'_{\pi_0}$ and $\mathbf{d}'_{\rho_0}$ as corresponding origins for the model and data features, respectively, we eliminated the translation component of the transformation. To find a full transformation we have to compute the translation component. Once we have found the optimal parameters for the matrix $\mathbf{S}$, we then place them back into the original constraints described in equation (5). For axial square uncertainty regions this results in a set of MN axial squares in the t_1-t_2 plane which can be analyzed in exactly the way the axial constraints in the ξ_1-ξ_2. In this way we can find the whole transformation $(\mathbf{S}, \mathbf{t})$.

The next section describes how, once we have found these candidate transformations we can then optionally refine them by more carefully enforcing the geometric constraints. The advantage here is that we have reduced the constraints on the 6 s-parameters to constraints on ξ_1 and ξ_2 which are much easier to analyze, yet still capture much of the geometric constraint.

5.3 Further Constraint Analysis

The previous sections have detailed a formulation in which constraints on feasible transformation parameters were projected from an 8D transformation parameter space down to a 2D parameter space, parameterized by ξ_1 and ξ_2. This is a conservative approximation in the sense that feasibility with respect to the original 8D constraints guarantees feasibility with respect to the projected constraints; the geometric consistency of a point in the transformation space after the approximation is always greater than or equal to the geometric consistency of the corresponding point before approximation.

The value of ξ_1 and ξ_2 found with this approximation can be taken as a starting point for a more localized exploration of the parameter constraint arrangement. To understand how this works we need to consider what the approximation actually does. This section develops this algebraically. In Appendix 8.3 we will derive this geometrically providing additional insight into the approach.

Once we have found candidates for ξ_1 and ξ_2 we can go back and refine our exploration of the constraint arrangement. Consider again equation (10) rewritten slightly:

$$c_i \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + [\, \tilde{\mathbf{d}}_{\pi_1} \quad \tilde{\mathbf{d}}_{\pi_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} - \tilde{\mathbf{d}}_j + [\, \mathbf{e}_{\pi_1} \quad \mathbf{e}_{\pi_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} = \mathbf{e}_j. \tag{18}$$

Earlier, to arrive at constraints on ξ_1 and ξ_2, we placed bounds on the error vectors in both the primary and secondary model features, $\mathbf{e}_{\pi_1}$, $\mathbf{e}_{\pi_2}$ and $\mathbf{e}_j$. Now we will refine our parameter estimation by exploring constraints on the primary data error vectors $\mathbf{e}_{\pi_1}$ and $\mathbf{e}_{\pi_2}$ when *given* an estimate for ξ_1 and ξ_2 from the first constraint exploration. To do this we will bound the error in the secondary error vector $\mathbf{e}_j$ only. This will now require $\mathbf{e}_{\pi_1}$ and $\mathbf{e}_{\pi_2}$, which affect the image location of all secondary model features, to have the same value for each reprojected model feature; this was not required before.

Specializing the general constraint inequalities again to the case of axial square uncertainty bounds leads to the constraints

$$|(c_i \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + [\, \tilde{\mathbf{d}}_{\pi_1} \quad \tilde{\mathbf{d}}_{\pi_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} - \tilde{\mathbf{d}}_j + [\, \mathbf{e}_{\pi_1} \quad \mathbf{e}_{\pi_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix})^t \hat{\mathbf{x}}| \leq 2\epsilon \tag{19}$$

$$|(c_i \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + [\, \tilde{\mathbf{d}}_{\pi_1} \quad \tilde{\mathbf{d}}_{\pi_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} - \tilde{\mathbf{d}}_j + [\, \mathbf{e}_{\pi_1} \quad \mathbf{e}_{\pi_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix})^t \hat{\mathbf{y}}| \leq 2\epsilon. \tag{20}$$

Consider, by way of explanation, the constraints in the $\hat{\mathbf{x}}$ direction. The inequality above becomes

$$(c_i\xi_1 + a_i\tilde{d^x_{\pi_1}} + b_i\tilde{d^x_{\pi_2}} - \tilde{d^x_j} + a_i e^x_{\pi_1} + b_i e^x_{\pi_2}) \leq 2\epsilon$$

and

$$(c_i\xi_1 + a_i\tilde{d^x_{\pi_1}} + b_i\tilde{d^x_{\pi_2}} - \tilde{d^x_j} + a_i e^x_{\pi_1} + b_i e^x_{\pi_2}) \geq -2\epsilon.$$

The only variables here are $e^x_{\pi_1}$ and $e^x_{\pi_2}$ and these two inequalities define two halfplanes in the $e^x_{\pi_1}$-$e^x_{\pi_2}$ plane whose intersection is a strip constraining the feasible values of $e^x_{\pi_1}$ and $e^x_{\pi_2}$ with respect

to the particular feature match $(\mathbf{m}_i, \tilde{\mathbf{d}}_j)$. The arrangement in the $e^x_{\pi_1}$-$e^x_{\pi_2}$ plane of these constraint strips for all secondary matches $(\mathbf{m}_i, \tilde{\mathbf{d}}_j)$ provides another constraint arrangement which we search for values of the two error parameters $e^x_{\pi_1}$ and $e^x_{\pi_2}$ yielding a transformation consistent for a large number of feature matches. We have an exactly analogous constraint arrangement in the $e^y_{\pi_1}$-$e^y_{\pi_2}$ plane constraining $e^y_{\pi_1}$ and $e^y_{\pi_2}$.

Let's build some intuition for what this refinement has done. To recount, we first bounded $\mathbf{e}_{\pi_1}$, $\mathbf{e}_{\pi_2}$, and $\mathbf{e}_j$ to derive constraints on ξ_1 and ξ_2 in inequalities (12). We then took an estimate of ξ_1 and ξ_2 and derived new constraints on $\mathbf{e}_{\pi_1}$ and $\mathbf{e}_{\pi_2}$ by bounding $\mathbf{e}_j$. The bounds given by inequalities (12) were quite loose. When two secondary matches $(\mathbf{m}_i, \tilde{\mathbf{d}}_j)$ and $(\mathbf{m}_k, \tilde{\mathbf{d}}_l)$ both have particular parameters values for ξ_1 and ξ_2 feasible with respect to these constraints, this means that at these values and for *some* value of $\mathbf{e}_{\pi_1}$ and $\mathbf{e}_{\pi_2}$, the match $(\mathbf{m}_i, \tilde{\mathbf{d}}_j)$ and the primary matches were all aligned within the uncertainty bound, and for some *possibly different* value of $\mathbf{e}_{\pi_1}$ and $\mathbf{e}_{\pi_2}$, the match $(\mathbf{m}_k, \tilde{\mathbf{d}}_l)$ and the primary matches were all aligned within the uncertainty bound. That is, the constraints (12) do not require that two different secondary matches are aligned by the same configuration of the primary matches (the primary match configuration is governed by the values of $\mathbf{e}_{\pi_1}$ and $\mathbf{e}_{\pi_2}$ which *adjust* the measured value of $\mathbf{d}_{\rho_1}$ and $\mathbf{d}_{\rho_2}$). Because the matching of all secondary matches is expressed in terms of the primary triple, only a single value of $\mathbf{e}_{\pi_1}$ and $\mathbf{e}_{\pi_2}$ is allowed for all matches. The refinement captured by (20) provides more constraint yielding a tighter estimation of the transformation parameters. Note that because $\mathbf{e}_j = (\mathbf{e}'_j - \mathbf{e}'_{\pi_0})$ we are still allowing a looser bound on $\mathbf{e}'_{\pi_0}$. Jacobs performed a similar search of the error parameters for geometric matching in the case of planar objects[27]. One way of viewing the algorithm we have described is that we determine the viewpoint of the 3D object given by ξ_1 and ξ_2 and then compute an equivalent planar object that projects the same in the image. We then explore the error space to find transformations better aligning the features. In the planar case the original parameters s_{ij} are related to the error parameters $\mathbf{e}_k$ by an affine transformation[13].

In the case of 2D objects and 3D motion as described in section 4.4 we can do a similar exploration of refined constraints by first finding an estimate of η_1 and η_2 and then deriving constraints on the two components of $\mathbf{e}_{\rho_1}$ by bounding $\mathbf{e}_j$ as above. The resulting arrangement is a set of axial rectangles in the plane, and is easy to explore.

5.4 A Tradeoff: Geometric Consistency vs. Algorithmic Complexity

There is an important tradeoff we made in the previous section between finding fully geometrically consistent match sets, and the nature of the algorithm required to do so. In effect we moved part of the computation away from the complex computational geometric problem of analyzing constraint arrangements in higher dimensional spaces, to a simple combinatorial search through triples of primary matches, leaving the much smaller and simpler computational geometric problem of analyzing planar arrangements. Analyzing arrangements is hard algorithmically, combinatorial searches are easy; so this is a good trade. This also allows us to incorporate other methods for reducing the computation required like grouping of primary data triples. For this reduction we traded somewhat relaxed geometric consistency constraints.

A match set is geometrically consistent if there exists a single transformation aligning the individual matches each to within the uncertainty bounds. The basic algorithm analyzing the arrangement of feasible transformations sets $\mathcal{F}^{ij}$ facilitates finding all maximal geometrically consistent

match sets by searching the arrangement but requires time $O(N^8M^8)$. Note that in this formulation the errors in the geometry of the data features are independent and have only local effect.

When using a triple of primary matches forming affine bases in which to represent the features as shown in the final variation in the previous section we have in effect incorporated the uncertainty in the primary data features into the uncertainty in the secondary data feature $\tilde{\mathbf{d}}'_j$ yielding new, larger bounds on the feasible transformation parameters for a given feature match $(\mathbf{m}_i, \tilde{\mathbf{d}}_j)$. The result is we need only search a 2D constraint arrangement, but the tradeoff is that this is an approximation because geometric consistency in the constraints on ξ_1 and ξ_2 with respect to these conservative bounds does not imply full geometric consistency. It is an approximation because it allows each secondary match to be feasible for possibly different values of the primary error vectors $\mathbf{e}'_{\rho_0}$, $\mathbf{e}'_{\rho_1}$, and $\mathbf{e}'_{\rho_2}$, while these can only take on a single value if we enforce global consistency. It is, however, a careful approximation and a *conservative estimate* of full geometric consistency. This allows us to define the notion of a conservative geometric consistency estimate.

6 Handling the Combinatorial Component

The main problem is to search through primary match triples until we find a *correct* one, which will be associated with a high geometric consistency estimate[20, 31]. Once we have found such a triple we can stop looking, unless there are multiple instances of the object in the data.

Consider exploring primary match triples in random order until we find a correct one. There are MN feature matches. If αM, $0 \leq \alpha \leq 1$, model features actually appear in the image there are αM *correct* feature matches. If we uniformly randomly choose primary match triples, the probability that we will choose a correct one is $q \approx \frac{\alpha^3 M^3}{M^3 N^3} = \frac{\alpha^3}{N^3}$. If we randomly choose k match pairs with replacement the probability $Q = (1 - P)$ that we have not found a correct one after k trials is $Q = (1-q)^k$. Taking logarithms and making a series approximation, to be sure that we have found at least one correct primary match triples with probability P we must examine $k \approx \frac{\log Q}{-q} = \frac{N^3}{\alpha^3}(-\log Q)$ primary match triples. For $P = 98\%$ and $\alpha = 1$ we need only consider at most $4N^3$ random match pairs, that is by randomly choosing $4N^3$ match pairs we will find a correct one with probability better that 98% (even better by choosing without replacement). The idea is to stop combinatorial exploration after we find a model instance, or to be more careful, when the probability is high that we have found all instances; strictly speaking we cannot guarantee completeness unless we explore the whole space. We thus have a notion of *probabilistic completeness*.

Feature grouping could be a powerful method for reducing the combinatorial search through primary match triples[28, 17, 32] using color, relative position, geometry, etc. to group feature matches into primary match triples. The use of grouping in the full constraint search problem is not obvious, while its use in the case of the combinatorial search is quite clear.

7 Testing the Method

The method was tested on both synthetic data and real image data, both for the cases of 2D object undergoing 3D motion using the η-plane variation, and 3D objects undergoing 3D motion using the ξ-plane variation. The primary objective for using synthetic data was to carefully control the variables such as sensor uncertainty and the correct feature correspondence in order to provide some

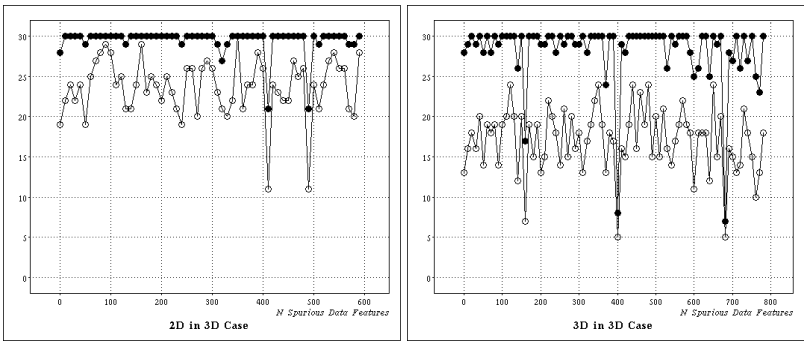

Figure 3: g1-graph-epsilon Results from synthetic tests. The curve with solid dots shows the number correctly matched features aligned in the image within 2ϵ, the curve with open dots shows the number correctly matched features aligned in the image within ϵ. The value plotted is the maximum, over three different correctly chosen primary match groups. Here $M = 30$, $\epsilon = 5$, the image size was 1024^2.

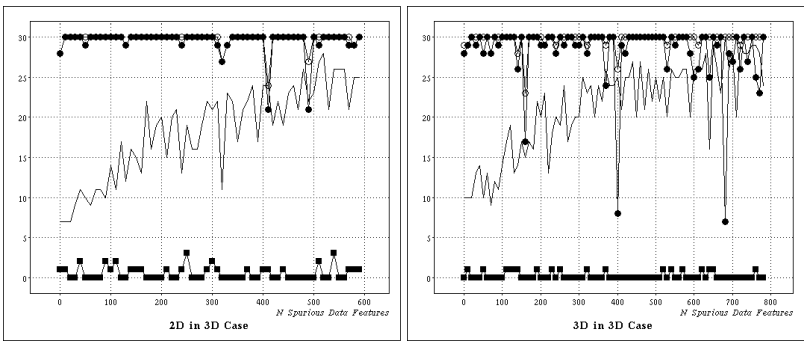

Figure 4: g1-graph-2epsilon Results from synthetic tests. The open dots show correct matches aligned within 2ϵ, the filled dots show the consistency measure for matches aligned within 2ϵ, in both cases for correct primary matches. The curve labeled with squares shows the number of correct matches aligned within 2ϵ, and the plain curve shows the consistency measure for matches aligned within 2ϵ, in both cases for incorrect primary matches. Here $M = 30$, $\epsilon = 5$, the image size was 1024^2.

ground truth for performance analysis. The size of the secondary feature match constraint squares in the ξ-plane and in the η-plane was computed using the Gaussian approximation method instead of the absolute bound as discussed in section 5.1.1 because the performance was better. This is demonstrated by synthetic experiments shown in figure 7.

7.1 Synthetic Data and Model Features

The synthetic model objects consisted of points generated uniformly in a circle in the 2D in 3D case, or a sphere in the 3D in 3D case, of about the size of the image dimensions. The models were imaged synthetically by applying a random transformation and projection to the image plane, then adding independent error to each point uniformly distributed over a square of side 2ϵ. Spurious data features were then generated uniformly in a circle the size of the image.

For the purposes of evaluation we will define some measures of geometric consistency. We define ξ-plane consistency as the geometric consistency with respect to the secondary match con-

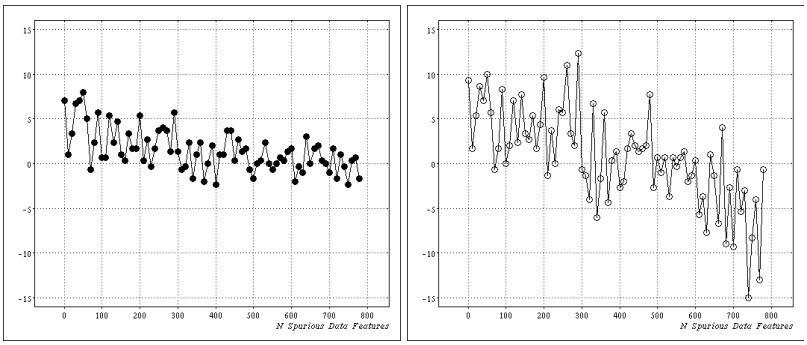

Figure 5: xi-alignment-graph On the left is shown the difference, in the number of correct matches aligned within 2ϵ, between the ξ-plane method and the alignment method. On the right is shown the difference, in the consistency measure for matches aligned within 2ϵ, between the ξ-plane method and the alignment method. Here $M = 30$, $\epsilon = 5$, the image size was 1024^2.

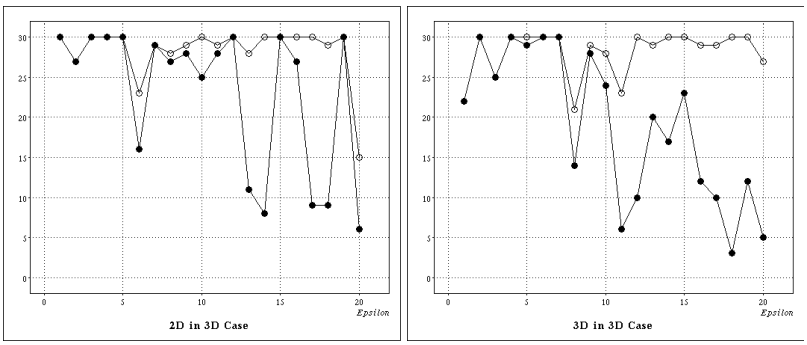

Figure 6: error-graph-ii As the error added to the data feature varies, along with the matching uncertainty bounds, the open dots show the consistency of matches aligned within 2ϵ in the image, the solid dots show the number of correct matches aligned within 2ϵ in the image. Here $M = 30$, ϵ varied from 0 to 20, and the image size was 1024^2.

straints in the ξ_1-ξ_2-plane. Similarly for the planar 3D case we define η-plane consistency. Define the ϵ-image-consistency as the geometric consistency of a transformation applied to the model features and then evaluated by comparison with the image data features and uncertainty constraints directly; that is, the consistency measure of the set of matches aligned to within ϵ by the transformation as describe in equation (1). Similarly we define 2ϵ-image-consistency as the consistency of the set of matches aligned to within 2ϵ in the image. Image consistency as defined here is a true measure of the goodness of a transformation–how well it matches the model to the image. Define ϵ-image-correct as the number of correct feature matches aligned within ϵ in the image, and similarly for 2ϵ-image-correct; for synthetic experiments we know which feature matches are correct so these can be computed.

The experimental results show that the method works well for finding transformations with high geometric consistency. The graphs in figure 3 each show two curves plotted for a synthetic model of size 30 features, and synthetic data containing additional spurious feature sets over a range of sizes. The graph on the left shows the 2D in 3D case using the η-plane method, the graph on the right shows the 3D in 3D case using the ξ-plane method. The 2ϵ-image-correct measure is plotted

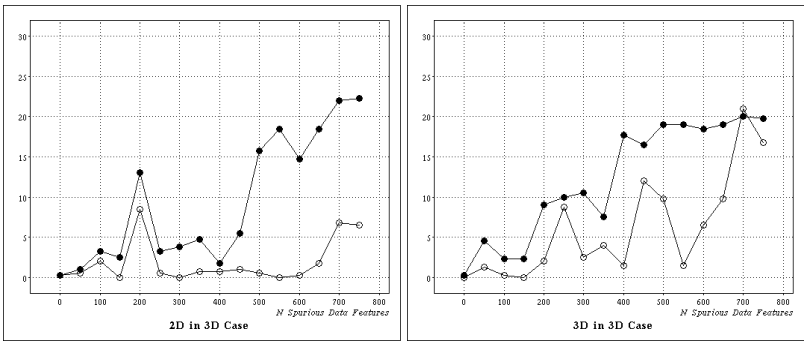

Figure 7: uniform-gaussian-graph Here is plotted the difference between the consistency of the matches aligned within 2ϵ in the image and the number of correct matches aligned within 2ϵ in the image, when using the absolute uncertainty bounds plotted as solid dots, and the Gaussian-derived approximation shown plotted as open dots. Here $M = 30$, $\epsilon = 5$, the image size was 1024^2.

as filled circles and the ϵ-image-correct measure is plotted as open circles, both for the case when the primary matches are correct. We can see that the transformations found are very close to correct ones, and align in the image most of the model features to within 2ϵ of their true corresponding data feature; a smaller fraction of matches are aligned to within ϵ. This shows that the projection to the ξ-plane, or η-plane, does provides a robust, yet approximate matching method: we're close to the optimal transformation, although not right on it.

Figure 4 illustrates how well the consistency measure approximates the number of correctly matched features. The figure shows two graphs; on left is the case of 2D objects in 3D using the η-plane method, on the right is the case of 3D objects in 3D using the ξ-plane method. The curve labeled with open circles is the 2ϵ-image-consistency measure, when the primary matches are correct, for the transformation constructed from the value of $\vec{\eta}$ or $\vec{\xi}$ found with the best η-plane-consistency or ξ-plane-consistency, as defined in equation (18). Plotted is the maximum over 3 random choices of correct primary matches to account for random bad choices of primary matches. The curves labeled with filled circles is the 2ϵ-image-correct measure for the same best transformation again maximized over three different primary match triples. From these two curves, which lie mostly on top of each other in the graph, we see that the consistency measure is close to the correct match count, which is 30 in this case.

It is interesting also to look at the consistency due to matches with spurious data by considering the case when the primary feature matches are incorrect. This is shown by the simple line curve in figure 4 plotted without hash marks which shows 2ϵ-image-consistency for the best transformation; the curve marked with squares shows the 2ϵ-image-correct count for incorrect primary matches. As we would expect, the image consistency of match sets consisting of incorrect, spurious matches grows in size with the number of spurious features–random alignments become more probable. The probability of such alignments is a function of the uncertainty region size, the image size, and the number of spurious features. Note that over a large range of spurious feature counts, up to a point where the number of spurious features is over 20 times that of actual model features, the false peaks due to incorrect primary matches are distinguishably lower than the peaks due to correct primary matches. Thus the ξ-plane consistency provides a reliable way to distinguish primary matches leading to high image consistency, and those which do not, provided the number of

spurious data features is not too high.

Figure 5 shows a comparison between the ξ-plane method and the basic alignment method for the 3D in 3D case. The figure shows the average difference in the 2ϵ-image-consistency and the 2ϵ-image-correct measures between the ξ-plane method and the standard alignment method. The open circles show the difference in the 2ϵ-image-correct measure, the filled circles show the difference in the 2ϵ-image-consistency measure. For moderate amounts of spurious data up to more than 10 times the model size, the ξ-plane method finds larger consistent or correct match sets. As the number of spurious features rises further, the ξ-plane method is lead away from the correct transformations. Note that the ξ-plane method allows non-rigid transformations, thus although the number of features matched is better than the alignment method, some of this is due to the broader set of possible transformations.

The behavior of the algorithm as the amount of error in the data is increased along with the uncertainty bounds in the algorithm is shown in figure 6. The open circles show the 2ϵ-image-consistency measure and the filled circles show 2ϵ-image-correct measure, in both cases when the primary matches are correct. As we would expect as the the uncertainty bounds grow relative to the image size the consistency measure remains the same, but the correlation between high consistency and high correctness decreases: there are more incorrect matches that have high consistency. This is also seen from the curve in figure 4 with no hash marks, where the primary matches are incorrect and the 2ϵ-image-consistency measure is plotted.

Figure 7 shows tests looking at the effects of the Gaussian approximation to the uniform uncertainty bounds described in section 5.1.1. The curve plotted with open circles shows the average, over 4 random correct primary matches, of the 2ϵ-image-correct when the Gaussian approximation to the η-plane or ξ-plane bounds were used. The curve plotted with filled circles shows the average, over 4 random correct primary matches, of the 2ϵ-image-correct when the absolute bounds were used. For higher numbers of spurious features the uniform bound leads us away from peaks associated with correct matches much more than the approximated bounds, because the tails of the distributions contribute more than they should in the case of absolute bounds.

7.2 Real Image Data

Tests of the method have also been performed on data derived from real images of both 2D and 3D objects. For 2D objects in 3D, images of signs were used to derive models and data directly from the images. Typically the model was derived from a perpendicular view of the planar sign, and the data from a different, arbitrary view. Figure 8 shows an example of an image from which the model was derived, as well as a data image; figure 9 shows both the data features, and the model features overlayed with the data features showing the best hypothesized matching. For a 3D object, two black metal bookends where fastened together to construct an interesting non-planar object. This object is shown in figure 10.

In the experiments we sought to eliminate the need for an initial feature grouping stage or a simple but expensive brute force search through primary feature matches, and focus on the constraint analysis phase. So, for each data image the model and data features where divided up into groups which might possibly be paired together. This was done by forming a rough alignment by hand, then grouping model features to data features which fall near each other. When running the algorithm, primary matches were chosen randomly from these groups; thus search through primary

Figure 8: An image, on the left, in which "xerox" was recognized using a model derived from a more or less perpendicular closeup, on the right, of the same sign.

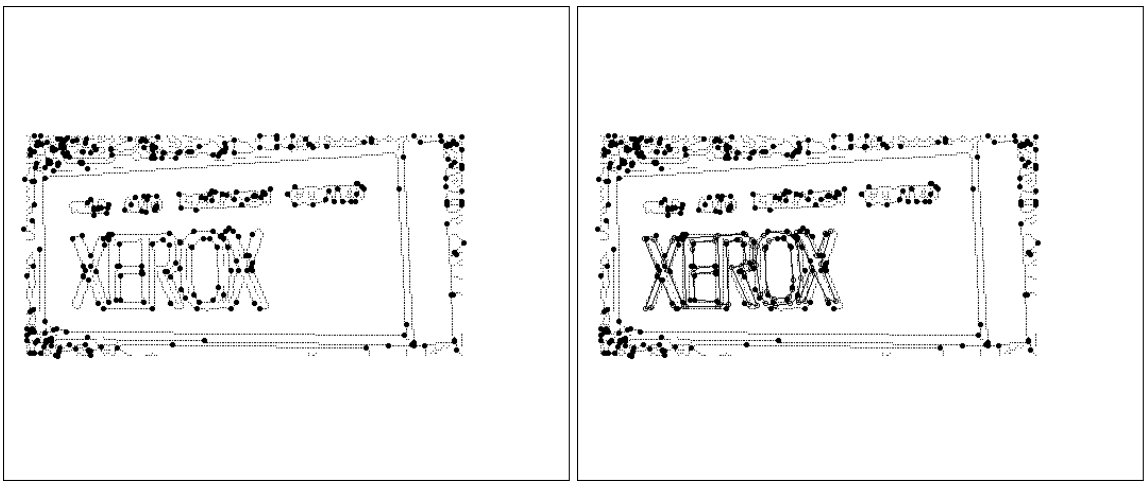

Figure 9: The data features, as black filled circles, and the data edges from which the were derived is shown on the left. On the right is the data with the best (highest consistency measure at 46/77) matching model transformation superimposed. The model size in this case was 77 features, the data size was 378 features. The uncertainty bound was 2 pixels, a 4x4 square region.

matches is still required but reduced in the computation required. The secondary matches were *not* chosen from these groups, rather all secondary matches were considered equally. Of course, we are possibly eliminating incorrect hypotheses with high geometric consistency that would normally be found, but nonetheless it does allow testing of the essence of the method.

Figures 11 and 12 show several different views of this object. On the left is shown the data features and the image edges from which they were derived; on the right is shown the same thing with the model features and edge segments overlayed. Shown is the hypothesis with the best ϵ-image-consistency measure. Sometimes the best value of this measure is not associated with a correct alignment. One cause of this is that we allow 3D affine transformations of the object. A second cause is that we have not brought to bear any orientation constraints in the ξ-plane analysis, although in this case with edge segments this is easily done[14]. Also, no verification has been performed, for example by comparing model edges and image edges, or the edge orientations associated with the point features which would eliminate completely implausible hypotheses. We simply wish to show here the performance of the hypothesis method.

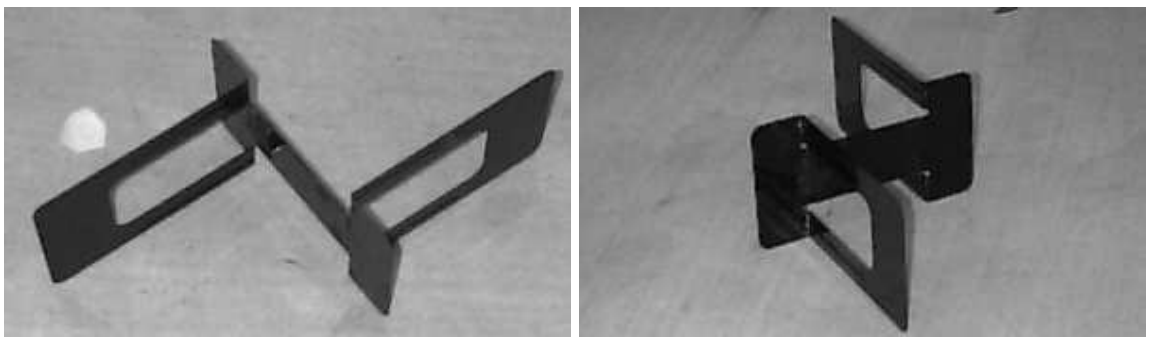

Figure 10: Two views of the object. The data were gathered from 8 such views. The model was built by hand measurements of this object.

8 Previous and Related Work

Perhaps the most similar geometric matching method to ours is the alignment method[25]. Interestingly the geometric constraint method and the alignment method have converged to a similar point as they have evolved. In the geometric constraint case it has moved from an 8D constraint analysis problem to a combined combinatorial search and a 2D geometric constraint analysis carefully handling the effects of uncertainty; in the case of alignment it has moved from mostly a combinatorial search through primary match triples to a combined combinatorial search and geometric constraint analysis problem[3] carefully accounting for the effects of data uncertainty.

Let's define the careful alignment method as the alignment method improved to account more carefully for the effects of uncertainty in the data features. Given three primary matches the alignment method computes the three-point pose solution. Next, to account for the effects of measurement uncertainty in the primary data features when reprojecting secondary model feature by the alignment transformation, the region in the image to which each model feature could reproject under all possible values of the error in the primary data features is either bounded or approximated[3, 2]. An easy way to see the effects of error on reprojection is from the reprojection equation for $\mathbf{m}_i$:

$$\mathbf{S}\mathbf{m}_i = c_i \begin{bmatrix} \tilde{\xi}_1 \\ \tilde{\xi}_2 \end{bmatrix} + [\, \tilde{\mathbf{d}}_{\pi_1} \quad \tilde{\mathbf{d}}_{\pi_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix}$$

where

$$[\, \tilde{\mathbf{d}}_{\pi_1} \quad \tilde{\mathbf{d}}_{\pi_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} = [\, \mathbf{d}_{\pi_1} \quad \mathbf{d}_{\pi_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} + [\, \mathbf{e}_{\pi_1} \quad \mathbf{e}_{\pi_2} \,] \begin{bmatrix} a_i \\ b_i \end{bmatrix} \tag{21}$$

and $\tilde{\xi}_1$ and $\tilde{\xi}_2$ have a non-linear dependence on the measurement error as seen from Appendix 10.1. One approach is to bound the possible region of reprojection, for example by making a first-order approximation to the reprojection equation[3] then bounding the error vectors. The data features whose uncertainty regions then intersect this region are potential matches.

One point of comparison between the careful alignment approach and the ξ-plane analysis approach is the constraint being applied. The ξ-plane approach does not enforce the rigidity constraints which constrain ξ_1 and ξ_2 with the two quadratic equations (see Appendix 10.1), however it does require all consistent matches to have *the same value* for ξ_1 and ξ_2: this is what the constraint analysis does. In contrast, the alignment procedure can allow different values of ξ_1 and ξ_2 for each

secondary match when analyzing the consistency of the hypothesis; this is not physically possible because all matches must all use one global value for these parameters.

Another point of comparison is the time complexity of the approaches. For the alignment approach in the case where our bounds are axial squares, using a 2D range query algorithm[33] we can determine how many of the N data point regions overlap the square reprojection region for each secondary model feature $\mathbf{m}_i$ in $O(\lg N + k)$ time using $O(N \lg N)$ space, where $k = O(N)$ is the number of points reported. This requires $O(N \lg N)$ preprocessing time up front given the data features. There are M model features, so evaluating the hypothesized primary match triple requires time $O(M \lg N + Mk)$ time, where k is typically a small fraction on N mostly due to spurious features, and might reasonably be neglected in practice. In contrast, for the geometric matching algorithm outlined in section 5 requires analysis of a 2D arrangement of MN axial squares requiring $O(MN \lg MN)$ time to determine the optimal number of matches aligned; the more sophisticated geometric consistency of equation (1) measure is only hueristically done.

In the case of planar objects the ξ-plane method is faster. It searches through $O(N^2M^2)$ primary match pairs and for each does the above $O(MN \lg MN) = O(MN \lg N)$ (for $M \leq N$) analysis to determine consistency for a total of $O(M^3N^3 \lg N)$. The alignment method searches through $O(N^3M^3)$ primary match triples and for each does an $O(M \lg N)$ analysis to determine consistency for a total of $O(N^3M^4 \lg N)$. Interestingly their expected-time performance by randomly choosing primary matches is the same: $O(N^3M \lg N)$.

In the case of non-planar objects the alignment method is faster, however to be accurate the ξ-plane approach is solving a harder problem: the affine structure matching problem. If we were to solve the affine matching problem using an alignment approach it would take $O(N^4M^5 \lg N)$ time by searching through prmary match *quadruples*. In the non-planar case the ξ-plane approach takes $O(N^4M^4 \lg N)$ time for the affine problem and the alignment approach takes $O(N^3M^4 \lg N)$ time for the rigid problem.

8.1 Other Matching Approaches

Several authors have described techniques based on constrained correspondence search[5, 22, 7, 2] where a hypothesized set of matches is used to bound the set of feasible transformation parameters. Correspondence space is searched in a way that prunes the search when the set of feasible transformations becomes empty. One variation uses the set of feasible transformations to bound the region to which a candidate model feature could reproject, in order to help choose the next feature match to add to the constraints[2]. We note, however, that with the exception of Breuel's approach[7] these still require an exponential search which is unnecessary, and was the main reason to explore the transformation constraint approach[13].

Another approach is based on transformation parameter clustering, such as Hough transform methods[6]. These approaches are very similar in spirit to our method, seeking to find regions of transformation constraints where many feature matches are consistent. An important difference is that it is difficult to quantize the space into buckets and carefully handle the effects of data feature error in transformation space. A related but more sophisticated approach has been called geometric hashing[31] which clusters transformation invariant parameters associated with the model object. This method can be used to find correspondences between data features and model features over a library of models. It essentially pre-compiles up front the search over primary model feature groups

so the combinatorial search is only performed over the data features. In addition to the difficulty of properly handling the effects of error in quantizing and bucketing methods in general, in the case of geometric hashing the effects of error on the invariant parameters, such as the 2D affine coordinates in the planar case, are even more complicated than in the standard parameters space this is discussed in more detail in a companion paper[15]. The effects of incorrect feature matches from multiple models also adds difficultly.

We have chosen a discrete, uniform error model for the data features to allow the straightforward use of methods from computational geometry. Other, richer error models have been considered such as using gaussian distributions to capture data feature error, in both matching[37] and geometric hashing[18, 34]. The absolute error bounds we use could be viewed as a simple, uniform probability distribution. They are a special case of a discrete, quantized approximation to continuous error distributions, which also incorporate the advantages of continuous, robust statistical methods by excluding outliers. Additionally, using these geometric approximations of continuous distributions allows us to guarantee the work required to find the solution, whereas in continuous optimization methods this can not often be guaranteed in practice.

8.2 Other Geometric Approaches

The method presented here evolved from an original approach to matching of oriented points in the plane using circular uncertainty bounds[9, 11] which essentially explored an arrangement of quadratic surfaces bounding feasible match parameters in the 3D space of rigid planar transformations to find feasible match sets. The essential point of that work was that the bounded matching problem was polynomial–not exponential as the robust matching algorithms of that period were. This essential idea has evolved to include higher dimensional feature sets and transformations[12, 13]. During this time the computational geometry community has become interested in versions of this problem[1], in particular a variation based on the Hausdorff distance[24, 36]. Two systems in particular are grounded in real recognition systems focused on 2D objects and affine transformations[23, 36].

8.3 The 6D Constraint Version

This section presents a more geometric view of the matching constraints which were derived algebraically in section 4. This geometric constraint approach was first done in the s-parameter space, parameterized by the components, s_{ij}, of $\mathbf{S}$[13, 14]. This is closely related to the approach presented here, which might be thought of as a careful approximation .

We will again assume the primary model features are in a canonical orientation so that $\mathbf{m}_{\pi_1}$ and $\mathbf{m}_{\pi_2}$ lie in the $z = 0$ plane. Essentially with this favorable model coordinate frame the first two columns of the s-parameter matrix $\mathbf{S}$, which defines a new matrix

$$\mathbf{S}_4 = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix}$$

with

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_4 & \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \end{bmatrix},$$

determine the transformation on the planar component of the secondary model features. Indeed, any mapping of the three primary model features to the image is captured by a planar affine transformation which is governed by the 4 parameters comprising the columns of $\mathbf{S}_4$, when the features are translated so that $\mathbf{d}'_{\rho_0}$ and $\mathbf{m}'_{\pi_0}$ are corresponding origins of the data and model coordinate frames, respectively. The last column of $\mathbf{S}$ can be seen to determine the mapping of the non-planar component of the secondary model features to the image. Note that the special orientation of the primary match triple is not required here, but simply makes the equations simpler and more intuitive. The general case of arbitrary model orientation is described in detail in a companion paper[15].

The algorithm described in section 4 essentially first explores the space of parameters ξ_1 and ξ_2 (s_{13} and s_{23} in the case of canonical model orientation) affecting the non-planar component of the model to find values with potentially high geometric consistency, then refines the parameter estimate by exploring the space of parameters s_{11}, s_{12}, s_{21}, and s_{22} affecting the planar component of the model to find values with high geometric consistency with respect to them. The algorithm of section 4 first found feasible values of ξ_1 and ξ_2 assuming the primary matches were loosely aligned, then found feasible values of the error vectors $\mathbf{e}_{\rho_1}$ and $\mathbf{e}_{\rho_2}$. As this section explains, this is analogous to finding feasible values of s_{13} and s_{23}, then fixing these values and finding feasible values for s_{11}, s_{12}, s_{21}, and s_{22}.

To build geometric intuition and make these relationships clear we will review the formulation of this problem described in Cass[13]. Consider again inequalities (5) for the primary base matches reproduced here.

$$\begin{aligned}(\mathbf{S}\mathbf{m}'_{\pi_0} + \mathbf{t} - \tilde{\mathbf{d}}'_{\rho_0})^t\hat{\mathbf{n}}_l &\geq -\epsilon^l \\ (\mathbf{S}\mathbf{m}'_{\pi_1} + \mathbf{t} - \tilde{\mathbf{d}}'_{\rho_1})^t\hat{\mathbf{n}}_l &\geq -\epsilon^l \\ (\mathbf{S}\mathbf{m}'_{\pi_2} + \mathbf{t} - \tilde{\mathbf{d}}'_{\rho_2})^t\hat{\mathbf{n}}_l &\geq -\epsilon^l.\end{aligned}$$

In the case of axial square uncertainty bounds we have $\hat{\mathbf{n}}_l \in \{\pm\hat{\mathbf{x}}, \pm\hat{\mathbf{y}}\}$ which allows us to combine these inequalities pairwise yielding 3 inequalities of the form

$$(\mathbf{S}\mathbf{m}_{\pi_{ij}} - \tilde{\mathbf{d}}_{\rho_{ij}})^t\hat{\mathbf{n}}_l \geq -2\epsilon^l$$

where $\mathbf{m}_{\pi_{ij}} = (\mathbf{m}'_{\pi_i} - \mathbf{m}'_{\pi_j})$ for $i \neq j$ and similarly for $\tilde{\mathbf{d}}_{\rho_{ij}}$. Because in this case $\hat{\mathbf{n}}_l \in \{\pm\hat{\mathbf{x}}, \pm\hat{\mathbf{y}}\}$ we get independent sets of constraints on the parameters (s_{11}, s_{12}, s_{13}) and on the parameters (s_{21}, s_{22}, s_{23}). For example consider the inequalities for $\hat{\mathbf{n}}_l = \pm\hat{\mathbf{x}}$ and $\epsilon^l = \epsilon$:

$$s_{11}m^x_{\pi_{ij}} + s_{12}m^y_{\pi_{ij}} + s_{13}m^z_{\pi_{ij}} \geq d^x_{\rho_{ij}} - 2\epsilon \tag{22}$$

$$s_{11}m^x_{\pi_{ij}} + s_{12}m^y_{\pi_{ij}} + s_{13}m^z_{\pi_{ij}} \leq d^x_{\rho_{ij}} + 2\epsilon. \tag{23}$$

Examination of inequalities (23) and (24) show that together they describe an infinite hexagonal tube in the 3D space parameterized by (s_{11}, s_{12}, s_{13}); similarly we have a *parallel* hexagonal tube in the 3D space parameterized by (s_{21}, s_{22}, s_{23}). This is easily seen because the normal vectors describing the planar constraints are proportional to the vectors $\mathbf{m}_{\pi_{ij}}$ for $i, j = 0, 1, 2, i \neq j$ which are coplanar; furthermore these constraints have a common solution because there is always a transformation mapping the primary model features onto the primary data features. Let's call these hexagonal tubes primary constraint tubes since they derive from the primary feature matches.

The constraints for a secondary match $(\mathbf{m}'_i, \tilde{\mathbf{d}}'_j)$ each take the form of infinite slices of the two 3D parameter spaces. Consider again the constraints on the parameters (s_{11}, s_{12}, s_{13}) when we take $\hat{\mathbf{n}}_l = \pm\hat{\mathbf{x}}$. The constraints take the form

$$[\mathbf{S}(\mathbf{m}'_i - \mathbf{m}'_{\pi_k}) - (\tilde{\mathbf{d}}'_j - \tilde{\mathbf{d}}'_{\rho_k})]^t(\pm\hat{\mathbf{x}}) \geq -2\epsilon \tag{24}$$

for $k = 0, 1, 2$. These constraints describe 3 planar slices of the 3D parameters space which intersect to form a polyhedron. This constraint polyhedron intersects the primary constraint tube to form a polyhedral region of parameters space for which the secondary match $(\mathbf{m}'_i, \tilde{\mathbf{d}}'_j)$ and the primary matches are all feasible. There is always an intersection because under an affine transformation and orthographic projection it is always possible to map, respectively, 4 model features onto 4 data features. If we consider the polyhedrons thus formed for each of the secondary matches, they form a constraint arrangement *inside* the primary constraint tube. This arrangement can be explored to find sets of parameters (s_{11}, s_{12}, s_{13}) with high geometric consistency. Note that we must explore the analogous, orthogonal constraint arrangement for the parameters (s_{21}, s_{22}, s_{23}); a fully consistent set of s-parameters is found from the cross product of these two arrangements.

Now, projecting this polyhedron of feasible parameters for match $(\mathbf{m}'_i, \tilde{\mathbf{d}}'_j)$ onto the spine of the hexagonal tube gives an *interval* along the spine. This interval is closely related to the set of feasible ξ_1 values for the secondary match $(\mathbf{m}'_i, \tilde{\mathbf{d}}'_j)$ developed in section 4. This is easy to see for the canonical model orientation because then s_{13} and s_{12} (ξ_1 and ξ_2 in general) parameterized the dimension along the the axis of the two primary constraint tubes. The arrangement of secondary match constraints, when projected to the tube spine, forms a 1D arrangement of feasible intervals for the secondary feature matches. The same is true for the orthogonal 3D parameter space parameterized by (s_{21}, s_{22}, s_{23}) where the spine of the tube is parameterized by ξ_2. We note that the constraints developed in section 4 are slightly looser than those developed here, but are closely related.

As before in section 5.1 we analyze a 2D arrangement of constraint squares in the ξ_1-ξ_2 plane. Finding a point in the ξ_1-ξ_2 plane of high consistency amounts to finding a point inside these two orthogonal hexagonal tube arrangements, along the spine of the tubes, of potentially high consistency. However the optimal point may lie inside the hexagonal bounds, but off the spine of the tube. The idea now is to explore these arrangements in these other dimensions, at the optimal ξ_1 and ξ_2, perpendicular to these dimension.

In section 5.3 we described how this initial estimate to the s-parameters, ξ_1 and ξ_2, is refined by exploring constraints on $e^x_{\rho_1}$ and $e^x_{\rho_2}$, in one 3D s-parameter subspace, and $e^y_{\rho_1}$ and $e^y_{\pi_2}$ in the other. We will now show that in the case of canonical model orientation, this is equivalent to exploring constraints on s_{11} and s_{12} in the (s_{11}, s_{12}, s_{13}) subspace and s_{21} and s_{22} in the (s_{21}, s_{22}, s_{23}) subspace.

From equation (29), and the definition $\tilde{\mathbf{d}}_j = \mathbf{d}_j + \mathbf{e}_j$ we see that there is a linear relationship between the s-parameters and the error in the primary data features:

$$\begin{bmatrix} s_{11} & s_{21} \\ s_{12} & s_{22} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{m}}^t_{\pi_1} \\ \bar{\mathbf{m}}^t_{\pi_2} \end{bmatrix}^{-1} \{ [\, \tilde{\mathbf{d}}_x \quad \tilde{\mathbf{d}}_y \,] - [\, \mathbf{e}^x_\rho \quad \mathbf{e}^y_\rho \,] \} \tag{25}$$

where

$$[\, \mathbf{e}^x_\rho \quad \mathbf{e}^y_\rho \,] = \begin{bmatrix} e^x_{\rho_1} & e^y_{\rho_1} \\ e^x_{\rho_2} & e^y_{\rho_2} \end{bmatrix} = \begin{bmatrix} \mathbf{e}^t_{\rho_1} \\ \mathbf{e}^t_{\rho_2} \end{bmatrix}. \tag{26}$$

The constraint arrangements we explored in section 5.3 were closely related to the constraint arrangement you get by slicing the 3D constraint arrangement in, say, the x subspace with a plane

perpendicular to the axis of the tube, at the point ξ_1. The result is a 2D arrangement of constraints in the s_{11}-s_{12} plane, which is related by a 2D affine transformation to the constraints on $e^x_{\rho_1}$ and $e^x_{\rho_2}$.

In summary, when the primary model triple is in canonical orientation, the parameters s_{11}, s_{12}, s_{21}, and s_{22} affect the mapping of the planar component of a secondary model feature $\mathbf{m}_i$, and for a given value of these parameters, s_{13} and s_{23} affect the non-planar component. To find a point in the s-parameter space $(s_{11}, s_{12}, s_{21}, s_{22} s_{13}, s_{23})$ with high geometric consistency we first align the primary matches and find the parameters s_{13} and s_{23} (ξ_1 and ξ_2) consistent for many secondary matches. This conservatively finds parameters consistent with respect to the *non-planar* components of the secondary model features. For these parameters values we then refine the parameter estimation by exploring one constraint arrangement for the parameters s_{11} and s_{12}, and a separate, orthogonal constraint arrangement for the parameters s_{21} and s_{22}, which finds places of high geometric consistency with respect to the *planar* component of the secondary model features.

9 Acknowledgements

The author thanks Tao Alter, David Jacobs, Dan Huttenlocher, and William Rucklidge for helpful comments and discussions.

10 Appendices

10.1 The Scaled-Orthographic Three-Point Pose Solution

The three primary model and data features are aligned exactly by a transformation $\mathbf{S}$ satisfying

$$\mathbf{S} \left[\begin{array}{cc} \mathbf{m}_{\pi_1} & \mathbf{m}_{\pi_2} \end{array} \right] = \left[\begin{array}{cc} \mathbf{d}_{\rho_1} & \mathbf{d}_{\rho_2} \end{array} \right]. \tag{27}$$

Assuming the primary matches are in cannonical orientation (see section 3.1) and provided $\mathbf{m}_{\pi_1}$ and $\mathbf{m}_{\pi_2}$, and thus $\bar{\mathbf{m}}_{\pi_1}$ and $\bar{\mathbf{m}}_{\pi_2}$ are linearly independent this leads to an expression for the first four unknown transformation parameters.

$$\begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} = \left[\begin{array}{cc} \mathbf{d}_{\rho_1} & \mathbf{d}_{\rho_2} \end{array} \right] \left[\begin{array}{cc} \bar{\mathbf{m}}_{\pi_1} & \bar{\mathbf{m}}_{\pi_2} \end{array} \right]^{-1}. \tag{28}$$

From the quadratic constraints described in equation (4) we have

$$\mathbf{S}\mathbf{S}^t = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix}^t + \begin{bmatrix} \xi_1^2 & \xi_1 \xi_2 \\ \xi_2 \xi_1 & \xi_2^2 \end{bmatrix} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$$

This leads to two hyperbolic curves constraining ξ_1 and ξ_2

$$\xi_1^2 + \mathbf{d}_x^t \mathbf{M}' \mathbf{d}_x = \xi_2^2 + \mathbf{d}_y^t \mathbf{M}' \mathbf{d}_y, \tag{29}$$

and

$$\xi_1 \xi_2 + \mathbf{d}_x^t \mathbf{M}' \mathbf{d}_y = 0. \tag{30}$$

where for convenience we have defined

$$[\,\mathbf{d}_x \quad \mathbf{d}_y\,] = \begin{bmatrix} \mathbf{d}_{\rho_1}^t \\ \mathbf{d}_{\rho_2}^t \end{bmatrix} \tag{31}$$

and

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_{\pi_1}^t \\ \mathbf{m}_{\pi_2}^t \end{bmatrix} [\,\mathbf{m}_{\pi_1} \quad \mathbf{m}_{\pi_2}\,] = \begin{bmatrix} \bar{\mathbf{m}}_{\pi_1}^t \\ \bar{\mathbf{m}}_{\pi_2}^t \end{bmatrix} [\,\bar{\mathbf{m}}_{\pi_1} \quad \bar{\mathbf{m}}_{\pi_2}\,]$$

where

$$\mathbf{M}' \equiv \mathbf{M}^{-1}$$

Eliminating one of ξ_1 or ξ_2 from these equations results in a simple biquadratic equation which can be solved to yield ξ_1 and ξ_2. This yields the solution for $\mathbf{S}$ corresponding to a rigid rotation and scaled orthographic projection of the three primary model features onto their corresponding primary data features.

The quadratic constraints on the two components ξ_1 and ξ_2 of the matrix

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & \xi_1 \\ s_{21} & s_{22} & \xi_2 \end{bmatrix}$$

arise when the matrix $\mathbf{S}$ represents a rigid 3D rotation and scaled projection. If we relax the quadratic constraints on ξ_1 and ξ_2 we simply have an arbitrary affine transformation and projection of the model features to the image plane.

10.1.1 Affine Coordinates and Model Indexing

This particular formulation of the alignment computation makes clear the relationship of alignment and affine model indexing as formulated by Jacobs[29]. While we would expect this relationship because in both case we are concerned with *views* of an object, the relationship is strikingly simple. Define the affine image coordinates[31] of a secondary image feature by $\mathbf{d}_j = [\,\mathbf{d}_{\pi_1} \quad \mathbf{d}_{\pi_2}\,] \begin{bmatrix} \alpha_j \\ \beta_j \end{bmatrix}$ or when $\mathbf{d}_{\pi_1}$ and $\mathbf{d}_{\pi_2}$ are independent $\begin{bmatrix} \alpha_j \\ \beta_j \end{bmatrix} = [\,\mathbf{d}_{\pi_1} \quad \mathbf{d}_{\pi_2}\,]^{-1}\,\mathbf{d}_j$. We can then relate a feature's affine model coordinates (a_j, b_j, c_j) to its affine image coordinates (α_j, β_j) using equation (8) to get $\begin{bmatrix} \alpha_j \\ \beta_j \end{bmatrix} = c_i \begin{bmatrix} \xi_1' \\ \xi_2' \end{bmatrix} + \begin{bmatrix} a_i \\ b_i \end{bmatrix}$ where $\begin{bmatrix} \xi_1' \\ \xi_2' \end{bmatrix} = [\,\mathbf{d}_{\pi_1} \quad \mathbf{d}_{\pi_2}\,]^{-1} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}$. An expression for the affine structure of the model following the approach of Jacobs[30, 28] then follows by collecting all the model features into vectors[15] and we have $\vec{\alpha} = \vec{a} + \xi_1 \vec{c}$ and $\vec{\beta} = \vec{b} + \xi_2 \vec{c}$. Note that there are only two parameters governing the view of the object.

The difficulty of working with affine image coordinates can be seen by considering the effects of error in the data features. In this case

$$\begin{bmatrix} \tilde{\alpha}_i \\ \tilde{\beta}_i \end{bmatrix} = \{[\,\mathbf{d}_{\pi_1} \quad \mathbf{d}_{\pi_2}\,] + [\,\mathbf{e}_{\pi_1} \quad \mathbf{e}_{\pi_2}\,]\}^{-1} (\mathbf{d}_i + \mathbf{e}_i) \tag{32}$$

where the effect of error in the two basis vectors is non-linear in the error vectors $\mathbf{e}_{\pi_1}$ and $\mathbf{e}_{\pi_2}$ because they appear in quadratic terms in the inverse matrix.

10.2 Algorithms for Axial Rectangles Coverage

The maximal coverage of a set of axial rectangles in the plane can be computed using a segment tree data structure and a standard sweep algorithm. This section describes extensions to the basic coverage algorithm[33] to compute a heuristic version of the more sophisticated measures of consistency, $\Phi_\xi(\xi_1, \xi_2)$ and $\Phi_\xi^H(\xi_1, \xi_2)$. We will assume familiarity with the basic coverage algorithm, and summarize it only briefly.

A segment tree is a log-depth data structure which dynamically maintains the number of intervals covering any subinterval in an arrangement of intervals, as intervals are sequentialy inserted or deleted from the tree. To find the coverage of axial rectangles we do a sweep of the rectangles in the $\hat{\mathbf{x}}$ direction. When we encounter the left edge of a rectangle, we add the interval corresponding to the rectangle's extent in the $\hat{\mathbf{y}}$ direction to the segment tree. When we encounter the right edge we remove the interval from the tree. The segment tree is built such that each node is associated with a contiguous range, and its subtrees recursively represent completely contained subintervals. Thus at any subtree the coverage of the associated set of subintervals in the arrangement is known. At any point in the sweep the maximal coverage for the intervals in every subtree is maintained and thus the maximal coverage of the currently active intervals in the tree is maintained at the root; maximizing this after every interval insertion finds the maximal coverage point.

In the usual rectangle coverage algorithm, each node v is associated with a range bounded by two interval start and end points from the set of intervals, and maintains two values $k(v)$ and $t(v)$. The number $k(v)$ is the count of intervals assigned to that node v, which are those intervals that fully cover the range associated with the node but are not assigned to any ancestor in the tree. The number $t(v)$ is the maximal coverage over all the subintervals contained by the range associated with v, which are assigned to it's descendants. When we assign an interval to a node v in the tree we increment the sum, $k(v)$ associated with that node, then set

$$t(v) = k(v) + \mathtt{max}(t(v_l), t(v_r))$$

where $t(v_l)$ and $t(v_r)$ are the maximal coverages for the left and right children of v, denoted v_l and v_r respectively. The effects of this change are then propagated up to the ancestors of v performing exactly the same update computation, and the root maintains the maximal coverage for the active intervals in the sweep. This is reversible when an interval is removed, and so the segment tree maintains the current maximal coverage at any point in the sweep.

In the modified algorithm to compute the maximum of $\Phi_\xi(\xi_1, \xi_2)$ over the arrangement of squares each interval is associated with either a model feature or a data feature.[4] The modified algorithm keeps several values at each node in the tree: $k^d(v)$, the number of data intervals assigned to the node, $k^m(v)$, the number of model intervals assigned to the node, $t^d(v)$ and $t^m(v)$ independently maintaining the maximal coverage of data and model intervals for the subtree at v, and $G^d(v)$ and $G^m(v)$ to be defined below. The maximum of $\Phi_\xi(\xi_1, \xi_2)$ is given by the maximum of $G(root) = \mathtt{min}(G^m(root), G^d(root))$ over the whole sweep.

Define $s^d(v)$ to be the number of data intervals assigned to ancestors of v, i.e. the data intervals above which cover all subintervals covered by the subtree at v in the tree; similarly let $s^m(v)$ to be the model coverage of all intervals above v in the tree. To update a node after insertion or deletion

[4] Because of the partitioning step described in Section 5.2 no two intervals labeled with the same model or data feature overlap, except possibly at their enpoints.

we first update $t^d(v)$ and $t^m(v)$. If

$$\texttt{min}(s^d(v)+k^d(v)+G^d(v_l), s^m(v)+k^m(v)+G^m(v_l)) > \texttt{min}(s^d(v)+k^d(v)+G^d(v_r), s^m(v)+k^m(v)+G^m(v_r))$$

then $G^d(v) = k^d(v)+G^d(v_l)$ and $G^m(v) = k^m(v)+G^m(v_l)$ otherwise $G^d(v) = k^d(v)+G^d(v_r)$ and $G^m(v) = k^m(v) + G^m(v_r)$. This is then done recursively up the tree to the root. This local update procedure does not always fully maintain the correct value of $G(root)$ because assigning an interval to a node v can change the values of $G^d(v_j)$ and $G^m(v_j)$ for descendants v_j of v. As a heuristic update procedure, after an interval is assigned to v we follow a single path to a leaf following branches at nodes v_i for which

$$\texttt{min}(s^d(v_i) + t^d(v_i), s^m(v_i) + t^m(v_i)) > G(root)$$

. This quantity is an upper bound for $G(root)$ computed only for paths to leaves through v_i. Ties for choosing between children of a node are broken by flipping a coin. When we reach a leaf we back up the tree performing the above local update. Because this log-depth walk to a leaf happens with every insertion or deletion, this has the effect of sometimes finding inconsistencies in the tree and fixing them. Note that the correct value of $G(root)$ is always higher than the heuristic value, so we are only computing a conservative estimate.

References

[1] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl. Congruence, similarity, and symmetries of geometric objects. In *Discrete and Computational Geometry*, volume 3, pages 237–256. Springer-Verlag, New York, 1988.

[2] T. D. Alter and David W. Jacobs. Error propagation in full 3d-from-2d object recognition. 1994.

[3] Tao D. Alter. Robust and efficient 3d recognition by alignment. Master's thesis, Massachusetts Institute of Technology, 1992.

[4] Tao D. Alter. 3-d pose from 3 points using weak-perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):802–808, 1994.

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

[6] Dana H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13:111–122, 1981.

[7] Thomas M. Breuel. An efficient correspondence based algorithm for 2d and 3d model based recognition. A.I. Memo No. 1259, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, October 1990.

[8] Todd A. Cass. Polynomial-time geometric matching for object recognition. In *Submitted for publication review.*

[9] Todd A. Cass. Parallel computation in model-based recognition. Master's thesis, Massachusetts Institute of Technology, 1988.

[10] Todd A. Cass. Feature matching for object localization in the presence of uncertainty. In *Proceedings of the International Conference on Computer Vision*, Osaka, Japan, 1990.

[11] Todd A. Cass. Feature matching for object localization in the presence of uncertainty. A.I. Memo No. 1133, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, May 1990.

[12] Todd A. Cass. Polynomial-time geometric matching for object recognition. In *Proceedings of the European Conference on Computer Vision*, 1992.

[13] Todd A. Cass. *Polynomial-Time Geometric Matching for Object Recognition*. PhD thesis, Massachusetts Institute of Technology, 1992.

[14] Todd A. Cass. Robust geometric matching for 3d object recognition. In *Proceedings Int. Conf. on Pattern Recognition*, pages A477–A482, Jerusalem, Israel, 1994.

[15] Todd A. Cass. The three-point pose solution under scaled-orthography. *Unpublished Manuscript*, June 1995.

[16] L. Chew and K. Kedem. Improvements on geometric pattern matching problems. In *SWAT LNCS 621*, pages 343–348, Las Vegas, 1992.

[17] D. T. Clemens and D. W. Jacobs. Space and time bounds on model indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1007–1018, 1991.

[18] M. Costa, R.M. Haralick, and L.G. Shapiro. Optimal affine-invariant point matching. In *Proceedings of the 6th Israel Conference on AI*, pages 35–61, 1990.

[19] Herbert Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, 1987.

[20] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[21] W. Eric L. Grimson, Daniel P. Huttenlocher, and David W. Jacobs. Affine matching with bounded sensor error: A study of geometric hashing & alignment. Technical Report A.I. Memo 1250, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1990.

[22] W. Eric L. Grimson and Tomas Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):469–482, July 1987.

[23] Michael J. Hopcroft. *A Geometrical Approach to Model-Based Vision*. PhD thesis, 1995.

[24] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

[25] D. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(7):195–212, 1990.

[26] Daniel P. Huttenlocher and Todd A. Cass. Measuring the quality of hypotheses in model-based recognition. In *Proceedings of the European Conference on Computer Vision*, Genova, Italy, May 1992.

[27] David W. Jacobs. Optimal matching of planar models in 3d scenes. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, Maui, Hawaii, June 1991.

[28] David W. Jacobs. *Recognizing 3-D Objects Using 2-D Images*. PhD thesis, Massachusetts Institute of Technology, 1992.

[29] David W. Jacobs. 2d images of 3d oriented points. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, New York, NY, June 1993.

[30] Jan J. Koenderink and A.J. van Doorn. Affine structure from motion. *J. Opt. Soc. Amer. A*, 8(2):377–385, 1991.

[31] Y. Lamdan and H. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proceedings of the International Conference on Computer Vision*, pages 238–249, 1988.

[32] Clark F. Olson. Probabalistic indexing for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):518–522, 1995.

[33] E. P. Preparata and M. Shamos. *Computational Geometry–An Introduction*. Springer-Verlag, New York, 1985.

[34] Isidore Rigoutsos and Robert Hummel. A bayesian approach to model matching with geometric hashing. *International Journal of Computer Vision*, 62(1):11–26, July 1995.

[35] L. Roberts. Machine perception of three-dimensional solid objects. In J. Tippett, editor, *Optical and Electr-optical Information Processing*. MIT Press, Cambridge, MA, 1966.

[36] William Rucklidge. Locating objects using the hausdorff distance. In *Proceedings of the International Conference on Computer Vision*, Cambridge, MA,, June 1995.

[37] III William M. Wells. Statistical object recognition. Technical Report AITR–1398, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, January 1993.

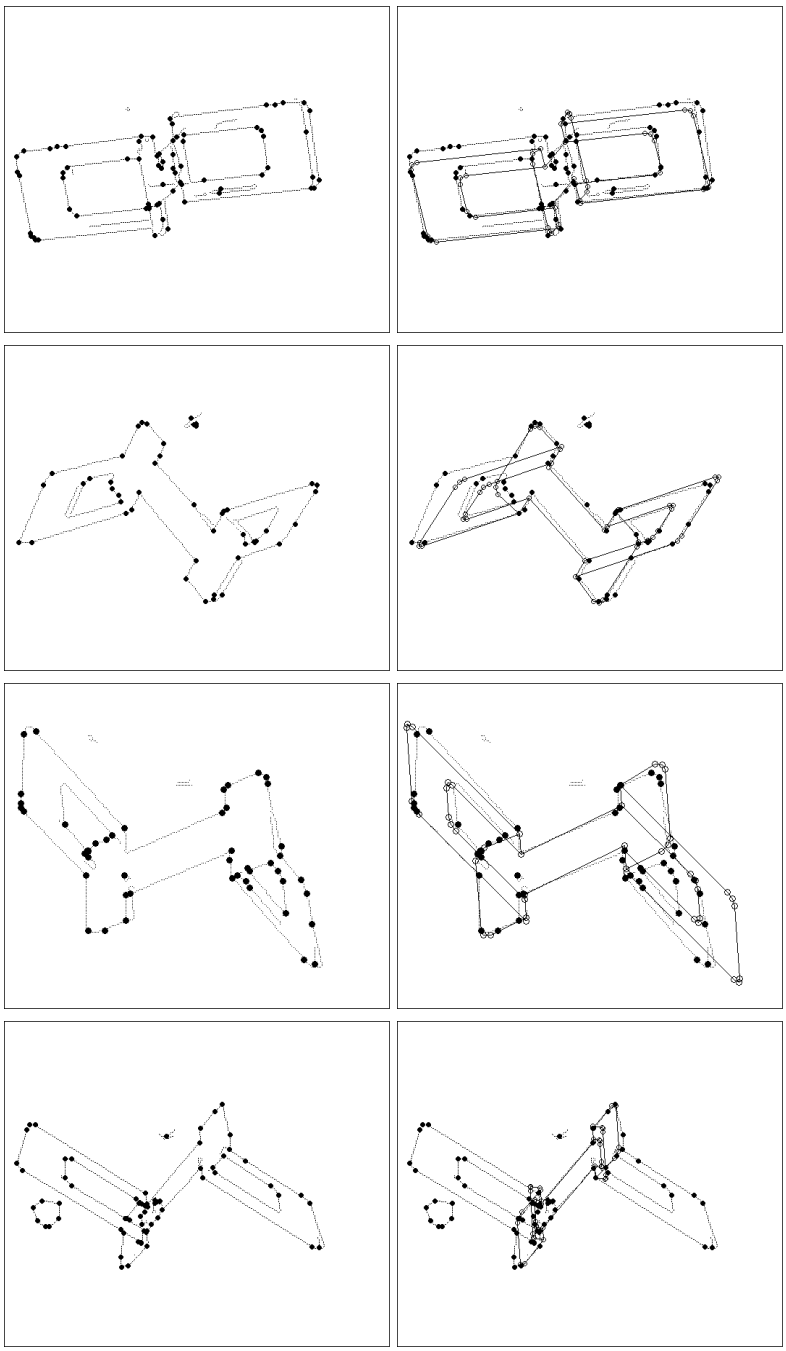

Figure 11: Experiments with real images of full 3D objects. On the left are the data features, connected by edge pixels. On the right is the best model transformation, shown by overlaying the model. The uncertainty allowed was 5 pixels, a 10x10 region.

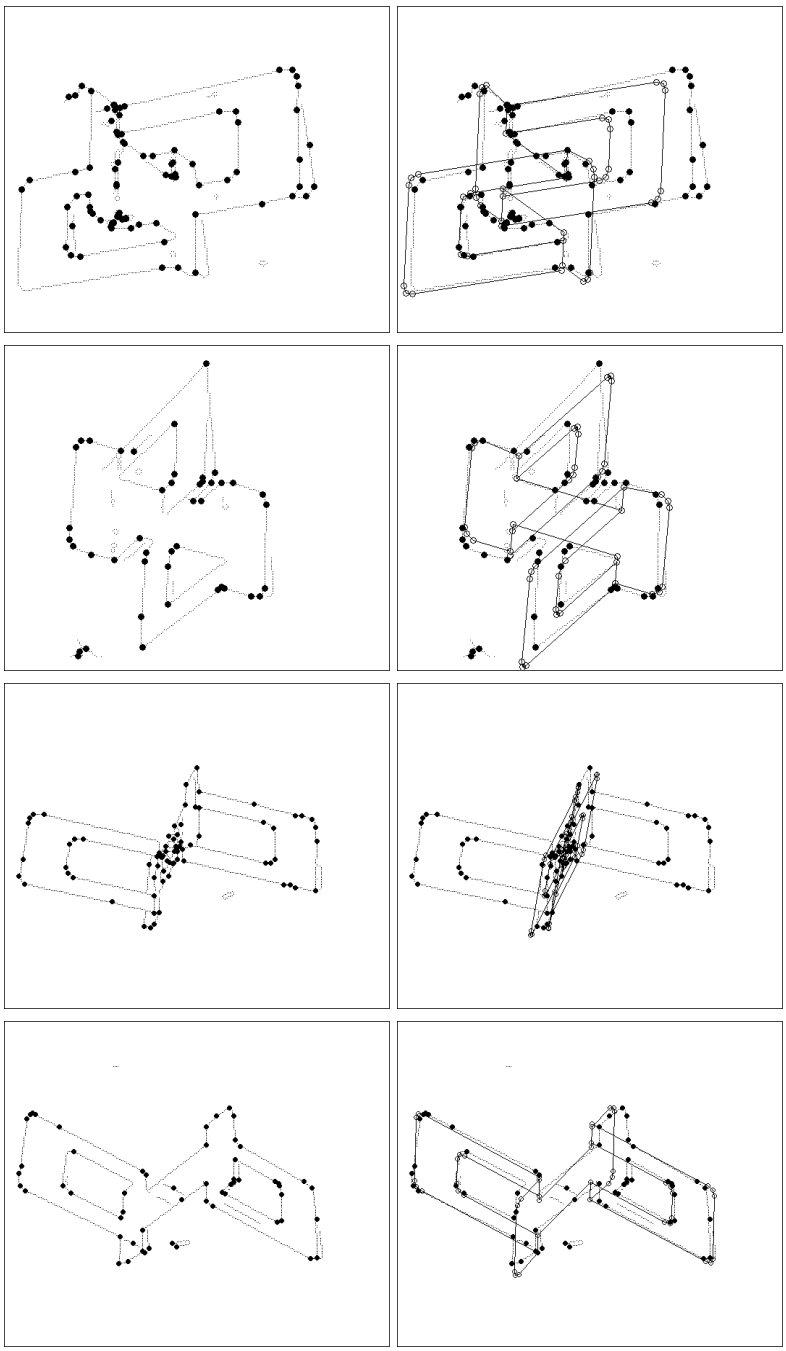

Figure 12: Experiments with real images of full 3D objects. On the left are the data features, connected by edge pixels. On the right is the best model transformation, shown by overlaying the model. The uncertainty allowed was 5 pixels, a 10x10 region.