



## 3-D to 2-D Pose Determination with Regions\*

DAVID JACOBS

*NEC Research Institute, 4 Independence Way, Princeton, NJ 08540, USA*

dwj@research.nj.nec.com

RONEN BASRI

*Department of Applied Math., The Weizmann Inst. of Science, Rehovot, 76100, Israel*

ronen@wisdom.weizmann.ac.il

*Received July 23, 1997; Revised September 4, 1998; Accepted March 30, 1999*

**Abstract.** This paper presents a novel approach to parts-based object recognition in the presence of occlusion. We focus on the problem of determining the pose of a 3-D object from a single 2-D image when convex parts of the object have been matched to corresponding regions in the image. We consider three types of occlusions: self-occlusion, occlusions whose locus is identified in the image, and completely arbitrary occlusions. We show that in the first two cases this is a convex optimization problem, derive efficient algorithms, and characterize their performance. For the last case, we prove that the problem of finding valid poses is computationally hard, but provide an efficient, approximate algorithm. This work generalizes our previous work on region-based object recognition, which focused on the case of planar models.

**Keywords:** object recognition, pose determination, linear programming, line traversal, occlusion, regions, parts, convexity

### 1. Introduction

Recognizing a known 3-D object using a single 2-D image is a central and difficult problem in visual recognition. One of the key issues is developing adequate representations to support flexible recognition of general objects. Existing approaches are often well-suited to only a small class of objects (e.g. polyhedra, rotationally symmetric objects, low-order algebraic surfaces). In this paper we show how to make use of a very simple and general representation of the parts of 3-D objects to determine their pose. At the same time, we also provide results on the fundamental computational

complexity of determining the pose of 3-D objects in the presence of occlusion.

A good representation for 3-D recognition should: (1) be rich enough to describe the shape of 3-D objects; (2) have a 2-D analog that can be reliably computed from an image; and (3) allow us to understand aspects of the relationship between the 3-D representation and its 2-D projection that are needed to perform useful recognition tasks. In this paper, we focus on the capability of a representation to support pose determination, one of the most basic problems faced by a complete recognition system.

#### 1.1. Past Work

Interesting methods have been developed for 3-D to 2-D recognition. However, the representations these methods use still have significant limitations. Many

\*A preliminary version of this paper has appeared as “3-D to 2-D Recognition with Regions” by D. Jacobs and R. Basri, IEEE Conf. on Computer Vision and Pattern Recognition, San Juan 1997:547–553. A brief overview of these and related results has appeared (Basri and Jacobs, 1996).

previous approaches have relied on finding a correspondence between simple geometric features, such as points or lines. Lowe (1985) and Clemens (1991), for example, determine pose based on a match between line segments, while Fischler and Bolles (1981), Huttenlocher and Ullman (1990), Horaud (1987), Ullman and Basri (1991), Jacobs (1997), Rothwell et al. (1993), and Alter and Jacobs (1998) use point features to determine pose, and Thompson and Mundy (1987) make use of vertices. It is fairly well understood how to use local features for pose determination or indexing, but they have significant weaknesses. Local features often do not capture the shape of complex, curved 3-D objects. And it may be quite difficult to locate 2-D image features that correspond to the local features of a non-polyhedral 3-D object, since the contour generator of such objects is completely viewpoint dependent.

There has been some recent work that extracts and matches point features from the outlines of smooth objects. Forsyth et al. (1992) use points derived from the bitangents of objects to derive an invariant description of the contour. This work, however, is limited to rotationally symmetric objects. Vijayakumar et al. (1995) show how to build an indexing function using bitangents for more general curved 3-D objects. They show the surprising result that a description based on bitangents can be represented as 1-D curves in a lookup table. These approaches use only a limited amount of the structural information available, however.

Another approach to recognizing smooth 3-D objects involves describing the 3-D object and 2-D image with algebraic surfaces and curves, and then registering these algebraic descriptions. Kriegman and Ponce (1990) have taken this approach, using elimination methods to solve for object pose. While this approach has provided significant insight into how the overall problem may be solved, it has the disadvantage of requiring a somewhat complex, iterative solution method. Specifically, their method requires a good estimate of pose to begin with, and then uses a variation of Newton's method to converge to the locally optimal pose. Forsyth (1996) has shown how to use an algebraic description of an image contour to determine the projective shape of the algebraic surface that produced it. This result is not practical, however, as it is extremely sensitive to noise. In general, while algebraic descriptions may be used to accurately represent a 3-D model, it is extremely difficult to derive a corresponding description of an image, since such descriptions may be very sensitive to noise.

A number of other approaches attempt to represent 3-D models using a specific vocabulary of shapes, typically based on an algebraic description, such as generalized cylinders (Binford, 1971; Brooks, 1981; Gross and Boulton, 1990; Marr and Nishihara, 1978; Ponce et al., 1989; Shafer and Kanade, 1983; Ulipinar and Nevatia, 1995; Zerroug and Nevatia, 1994), superquadrics (Pentland, 1987; Rivlin et al., 1995; Solina and Bajcsy, 1990; Terzopoulos and Metaxas, 1991), and geons (Bergevin and Levine, 1993; Biederman, 1985). Often these approaches handle only a limited class of objects. For example, when generalized cylinders are used a major difficulty lies in computing the 2-D projection of the 3-D axis and sweeping rule. Image occlusion and noise make this problem especially difficult. Considerable effort has led to solutions of this problem for only some restricted classes of shapes.

Moment-based methods are somewhat related to ours, in that they compute a description of image regions to match to model volumes. These methods might align regions based on their center of mass, or on higher order moments. Examples of this approach can be found in (Dudani et al., 1977; Hu, 1962; Nayar and Bolle, 1996; Nagao and Grimson, 1994; Persoon and Fu, 1977; Reeves et al., 1984; Richard and Hemami, 1974; Sadjadi and Hall, 1980). These methods do not extend to the recognition of a 3-D object from a single 2-D image, however. First of all, volumes of 3-D points always produce self-occlusion, since different subsets of the surface of the volume are visible from different viewpoints. Therefore, the center of mass of the projection of a model volume will not be the projection of the center of mass of the 3-D point set. Second, the center of mass of a surface that is curved in 3-D does not project to the center of mass of its image, even when there is no self-occlusion. This is because the extent to which different portions of the 3-D surface are foreshortened will depend on the viewing direction. A final disadvantage of methods based on moments is that they are sensitive to occlusion by other objects in the scene.

In summary, previous research has shown the difficulty of finding general representations to support object recognition. Representations capable of describing very large classes of objects have proven difficult to use. Representations that facilitate tasks such as pose determination are often restricted to particular classes of objects, such as polyhedral, planar or rotationally symmetric objects.

## 1.2. Our Approach

We address the problem of 3-D recognition from a 2-D image in a parts-based framework. That is, like most work on the recognition of general, curved 3-D objects, we divide a 3-D object into its component parts, and expect that a bottom-up grouping system will identify image regions that are candidate matches for these parts. We use a simple, direct representation of an object's parts as general volumes in 3-D, using 2-D areas to represent their image projection. This representation can be applied to any 3-D shape (we discuss restrictions concerning convexity later), and derived from a 2-D image without any need to fit algebraic constructs (e.g., conics, lines, corners) to parts located in an image, or to compute other intermediate properties of parts such as their axes. Therefore, our first two goals for a representation are satisfied very generally, with no assumptions except those imposed by a parts-based approach to recognition.

The bulk of this paper will attack the third goal of a representation by showing how we can relate 3-D volumes to 2-D regions for the important problem of pose determination. This extends our previous results, which focused on planar objects (Basri and Jacobs, 1997). At the same time, we use this general representation to show novel results about the fundamental complexity of pose determination for general 3-D objects. To do this, we divide the possible types of occlusions that may make pose determination difficult into three conditions. First, we consider self-occlusions. We provide a simple algorithm that uses linear programming to find pose, given correspondences between 3-D volumes and 2-D regions. This is the same method that we applied to planar objects in (Basri and Jacobs, 1997), although we provide some new results demonstrating when this algorithm will produce correct results. Second, we consider occlusions whose position has been identified in the image. That is, we assume that each edge bounding an image region is labeled as either a region boundary, or an occlusion boundary. We show that a variation on our original algorithm can handle this case too. Third, we consider arbitrary occlusions of unknown location. That is, we assume that a 2-D region is a subset of the projection of the corresponding 3-D volume, but that any of the boundaries of the region may be due to occlusion. We show that the problem of pose determination in this case is fundamentally harder, by showing that this is a superset of a problem in computational geometry that is known to be hard. We

stress that this result is not specific to our algorithms. It shows that for parts-based object recognition, the problem of pose determination is provably more difficult when occlusions are of unknown location in an image than when their position is known. This tells us, for example, that many locally optimal poses can exist. This can present problems for gradient descent algorithms. However, we then provide an approximate algorithm, which is computationally efficient, and show that in a number of cases this leads to accurate results. Finally, we show how to handle degenerate solutions, which can especially occur with this approximate solution.

Our choice of representation is also general in that it can include as a subset several commonly used local geometric features, such as points, corners, line segments, or lines. Each of these features can be used by our methods as a 3-D volume and a corresponding 2-D region. Therefore, when local features are present in an object, these can be integrated into a common algorithm with extended features.

While we focus on the pose determination problem, we expect these results to fit into a complete recognition system as follows. At compile time, we divide an object up into component parts, preferably convex. At run time, we use a grouping system to identify candidate image groups. One might use intensity-based segmentation or a system that finds salient convex sets of edges (Jacobs, 1996). We then consider matches between image groups and model parts, with a search that may be directed with the addition of cues such as color, as was done by Nayar and Bolle (1996). Pose is determined using our current work, and then a hypothetical projection of the model may be confirmed or rejected using additional cues. The steps of this process are illustrated in numerous experiments described in Jacobs (1992). That system robustly matched convex object parts, but used a feature-based indexing system not suitable for non-polyhedral 3-D objects. In our current paper, all of these steps are also implemented, except we do not experiment with search to match object parts, focusing instead on determining the capability of our system to produce accurate poses once the correct match is found.

In Section 2 we describe algorithms for determining pose in the case of self-occlusion alone (Section 2.1), occlusion of known locus (Section 2.2), and arbitrary occlusion (Section 2.3). Section 2.3 shows that arbitrary occlusion is difficult, and describes an approximate algorithm. In Section 3 we describe how much information each algorithm needs to uniquely determine

pose. Section 4 describes how to recover from degenerate solutions, and Section 5 presents our experiments.

## 2. Using Volumes and Regions to Determine Pose

We assume that a hypothesized match exists between a set of model volumes and image regions. Our goal is to use this match to determine the pose of the model. We will assume these volumes and regions are convex throughout most of this section, and discuss issues related to non-convexity at the end.

We assume that the model consists of a set of 3-D volumes denoted:  $V_1, \dots, V_k \subset \mathcal{R}^3$ , where each volume is an arbitrary subset of  $\mathcal{R}^3$ . Similarly, we assume that the image consists of 2-D regions, which are each subsets of  $\mathcal{R}^2$ , and which we denote by:  $R_1, \dots, R_k \subset \mathcal{R}^2$ . Our solution methods will apply to the case where the model and image sets are convex; if we wish to make use of non-convex volumes or regions we should first take their convex hulls. This means that our methods can naturally apply also when some or all of the correspondences are between point features, or (possibly partially occluded) line segments, since these are convex.

Next, we suppose that the image was generated by applying some transformation,  $T$ , that maps points in the 3-D model to points in the image. For the most part we will assume that this is a 3-D to 2-D affine transformation. We denote a point in model space by  $\vec{p} = (x, y, z)$  and in image space by  $\vec{q} = (u, v)$ . If  $\vec{q} = T(\vec{p})$  then we denote  $u = T_u(\vec{p})$  and  $v = T_v(\vec{p})$ . Our goal is to identify the transformation that will best explain the image regions as the product of their corresponding model regions.

As we point out in (Basri and Jacobs, 1997), the problem of finding a transformation that perfectly matches a set of model volumes to their corresponding image regions is a non-convex optimization problem. This follows from the fact that the set of feasible transformations need not be convex, or even connected. Consider for example the case of a model square matched to an identical image square. Matching the model exactly to the image can be performed in four ways (separated from each other by a 90° rotation). Obviously, no intermediate transformation provides a solution to this matching problem. While non-convex optimization problems are often attacked using tools such as gradient descent, we instead take the approach of showing that different problem formulations can make the matching process convex, and therefore easier

to optimize. In Section 3 we discuss the conditions under which these formulations will produce the correct model pose.

### 2.1. The Forward Constraints: Self-occlusion

First, we show that our problem becomes convex if we merely require that every model point projects inside the corresponding image region, by reviewing results that we have shown in (Basri and Jacobs, 1997). Formally, the *forward constraints* are satisfied by the transformation,  $T$ , if and only if  $\forall \vec{p} \in V_i, T\vec{p} \in R_i$  (that is,  $TV_i \subseteq R_i$ ). These constraints allow a volume to occlude itself (i.e., two model points may project to the same image point). They do not capture all possible constraint, however, since they do not require that each image point be explained by a corresponding model point.

We first consider a projection model consisting of a 3-D affine transformation followed by an orthographic projection, then we consider perspective projection. Denote the linear part of  $T$  by  $A$ , where  $A$  is a non-singular  $2 \times 3$  matrix with elements  $t_{ij}$ , and the translation part by  $\vec{t} = (t_x, t_y)$ . Then:

$$\begin{aligned} u &= t_{11}x + t_{12}y + t_{13}z + t_x \\ v &= t_{21}x + t_{22}y + t_{23}z + t_y. \end{aligned} \quad (1)$$

This projection model and its equivalent has been recently used by a number of researchers (Lamdan and Wolfson, 1988; Ullman and Basri, 1991; Koenderink and van Doorn, 1991; Tomasi and Kanade, 1992; Jacobs, 1997). It is also equivalent to applying scaled orthographic projection followed by a 2-D affine transformation (Jacobs, 1997), that is, taking a picture of a picture. Alternately, it is equivalent to a paraperspective projection followed by translation (Basri, 1996), where paraperspective is a first-order approximation to perspective projection (Poelman and Kanade, 1997; Sugimoto, 1996).

To express the forward constraints, we note that since  $R$  is convex, there exists a set of lines  $L_R$  bounding  $R$  from all directions such that for every point  $\vec{q} \in R$  and for every line  $l \in L_R$  we can write

$$l(\vec{q}) \geq 0. \quad (2)$$

Let the line  $l$  be expressed by the equation:  $Au + Bv + C \geq 0$ , where  $(A, B)$  is the unit vector normal to the line. Then the constraint  $TV \subseteq R$  can be written as

follows. Every point  $\vec{p} \in V$  should be mapped by  $T$  to some point  $\vec{q} \in R$ , and so

$$A(t_{11}x + t_{12}y + t_{13}z + t_x) + B(t_{21}x + t_{22}y + t_{23}z + t_y) + C \geq 0. \quad (3)$$

This constraint is linear in the transformation parameters. Denote

$$\vec{w}^T = (t_{11}, t_{12}, t_{13}, t_x, t_{21}, t_{22}, t_{23}, t_y)$$

the vector of unknown transformation parameters, and

$$\vec{g}^T = (Ax, Ay, Az, A, Bx, By, Bz, B).$$

We can rewrite the forward constraints as

$$\vec{g}^T \vec{w} \geq -C. \quad (4)$$

We can similarly handle affine transformations followed by perspective projection. In that case

$$u = \frac{f(t_{11}x + t_{12}y + t_{13}z + t_x)}{t_{31}x + t_{32}y + t_{33}z + t_z}$$

$$v = \frac{f(t_{21}x + t_{22}y + t_{23}z + t_y)}{t_{31}x + t_{32}y + t_{33}z + t_z}$$

where  $f$  is the focal length. The forward constraint  $Au + Bv + C \geq 0$  now contains the term  $t_{31}x + t_{32}y + t_{33}z + t_z$  in the denominator. This term must be positive since we require the object to appear in front of the camera. So, we can multiply both sides of the inequality by this term, again obtaining a linear constraint with the same general form as Eq. (4), with different definitions of  $\vec{w}$  and  $\vec{g}$ .

The set of forward constraints consists of all such constraints obtained for each pairing of a bounding line  $l \in L_R$  and a model point  $\vec{p} \in V$ . This gives us one constraint for every point in the model volumes and for every line tangent to the image regions. For curved objects, therefore, the number of constraints is infinite, but we may sample them as accurately as we desire. The issue of sampling is addressed in (Basri and Jacobs, 1995). For polyhedral volumes and polygonal regions the number of independent constraints is finite, and given by the product of the number of vertices of the model volumes and the sides of the image regions. The rest of the constraints are redundant.

We therefore seek a set of transformation parameters that satisfy a set of linear constraints of the form:

$$\vec{g}_i^T \vec{w} \geq c_i, \quad i = 1, \dots, n. \quad (5)$$

This can be written in matrix notation as:

$$G\vec{w} \geq \vec{c}. \quad (6)$$

We may find a set of parameters that satisfy these linear constraints using linear programming. To do this, we must also specify a linear objective function. A common way of doing this is by introducing an additional unknown,  $\lambda$ , in the following way.

$$\begin{aligned} & \max \lambda \\ \text{s.t. } & G\vec{w} \geq \vec{c} + \lambda \vec{1} \end{aligned} \quad (7)$$

A solution to Eq. (6) exists if and only if a solution to Eq. (7) with  $\lambda \geq 0$  exists. (Note that other objective functions, e.g., the perceptron function, can be used for recovering  $\vec{w}$ , see e.g., (Duda and Hart, 1973) for a discussion of solutions to the linear discriminant functions problem.)

When  $\lambda \geq 0$  its value represents the minimal distance of a point to any line bounding the region (Fig. 1). Maximizing  $\lambda$  amounts to attempting to contract the model volume inside the image region as much as possible. When  $\lambda < 0$  this attempt fails. In this case any model point that violates the constraints is mapped to a distance of no more than  $|\lambda|$  from a line bounding its target regions.

Enforcing the forward constraints is therefore very similar to finding a transformation that minimizes the directed Hausdorff distance,  $h$ , from the transformed model volume to the image region. This can be

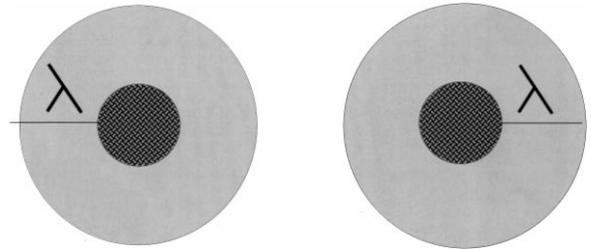


Figure 1. The dark circles are positioned by the similarity transformation that maximizes  $\lambda$  relative to the larger, shaded circles.

defined as:

$$h(M, I) = \max_{m \in M} \min_{i \in I} \|m - i\|$$

Hausdorff matching has previously been effectively used in computer vision by, for example, Huttenlocher et al. (1993a, 1993b), and Rucklidge (1997). These papers primarily focus on the use of robust variants of the undirected Hausdorff distance, which is defined as  $\max(h(M, I), h(I, M))$ , for aligning objects undergoing 2-D transformations. Amenta (1994) specifically discusses the efficient Hausdorff matching of convex shapes undergoing translation and scaling. While related to these approaches, our work is different in that we demonstrate new, efficient algorithms that compute something similar to the directed Hausdorff distance. These algorithms compute the pose of 3-D objects in single 2-D images.

More specifically, solving the system Eq. (7) may result in over-contraction. Consider, for example, the case of matching a single volume  $V$  to a single region  $R$ . The forward constraints restrict the set of possible transformations to those that map every point  $\vec{p} \in V$  inside the region  $R$ . Assume  $T$  is a feasible transformation, that is  $TV \subseteq R$ , then applying any contracting factor  $0 \leq s \leq 1$  to  $V$  would also generate a valid solution; namely,  $T(sV) \subseteq R$ . (We assume here without the loss of generality that the origin of the model is set at the centroid of  $V$ .) Consequently, the case of matching one volume with one region necessarily introduces multiple solutions. The solution picked by Eq. (7) is the one with  $s = 0$ . This will contract  $V$  to a point, which is then translated to the point inside  $R$  furthest from any of its bounding tangent lines. This solution produces the largest value of  $\lambda$ . Clearly, the case of matching one volume to one region cannot be solved by the forward constraints alone. However, we will show that we can determine pose accurately when we match a larger number of volumes and regions.

## 2.2. *The Forward Constraints with Known Occlusion*

The forward constraints are suitable when there is only self-occlusion; the backward constraints, as we will discuss, hold in the presence of arbitrary occlusion. We now consider one other possibility, that some model volumes may be partially occluded in the image, but that the extent of this occlusion is known. That is, we assume that some portion of boundary of the image

regions may be caused by an occluding object in front of the volume, rather than the boundary of the volume itself, but that we have identified those portions of the region boundaries. This is a situation of considerable interest; for example, if we identify several different parts of an object we may be able to determine that some of these parts lie in front of and occlude others (e.g., by identifying concave sections in the boundaries of a region corresponding to a convex volume). Also, it is well-known in the psychology literature that the knowledge that certain boundaries are due to occlusions can greatly assist human perception (Rock, 1983). Suppose we have identified a region in the image, but we know that some of the boundary of this region is due to another, occluding object and is not in fact the boundary of the region itself. By allowing for the region to be extended in the direction of such occlusions, we can construct a larger convex region which we know should contain the projection of the corresponding model volume.

This can be implemented with very little modification to the above algorithm. Suppose we approximate the border of a detected region with a set of line segments, some of which we know come from the boundary of the region, and some of which come from an occlusion. We apply the forward constraints, as described above, using only those line segments that originate due to the boundary of the object. Implicitly, this restricts the model volume to project within the largest possible convex region that is consistent with the known region boundary.

## 2.3. *The Backward Constraints: Unknown Occlusions*

We can allow for arbitrary, unidentified occlusions in the image with the *backward constraints*. The backward constraints are the inverse of the forward. Instead of requiring that each model volume project completely inside each corresponding image region, the backward constraints require that each image region lie completely inside the projection of each corresponding model volume. That is, the backward constraints are satisfied by the transformation,  $T$ , if and only if for every point,  $\vec{q}$ , in every image region,  $\vec{q} \in TV_i$  (that is,  $R_i \subseteq TV_i$ ).

It is preferable to enforce the backward constraints, rather than the forward constraints, when recognizing objects in the presence of occlusion. Typically, when a model volume is occluded by a different object or a

different model volume, a region detector will locate image regions that are subsets of the true, unoccluded regions. In this case, the true transformation may map a model volume so that part of the volume accounts for the visible portion of the region, and part of the model volume is mapped outside this region. Therefore, the backward constraints express precisely our state of knowledge about the pose when we allow for arbitrary occlusion of the image regions.

As we show in (Basri and Jacobs, 1997), in the case of *planar* objects we may solve the backward constraints using linear programming. This is because the transformations that we use to relate a planar model to a planar image (similarity, affine or projective) are invertible. Therefore, requiring that the viewing transformation maps the model volumes so that they completely enclose the corresponding image regions is equivalent to requiring that the inverse transformation map the image regions completely inside the corresponding model regions. For planar models, the forward and backward constraints have an identical form, with the role of the image and model reversed, and so we may find poses that satisfy them using the same method.

In the case of a 3-D model and a 2-D image, the 3-D to 2-D viewing transformation is no longer invertible, however. It is therefore not obvious how one can apply the backward constraints in this case.

In fact, we can prove that it is not possible to directly extend our planar results on the backward constraints to 3-D models. To do this, we now show that there is a close connection between the problem of solving the backward constraints and the computational geometry problem of finding a *line traversal*, i.e., finding a line that intersects a set of 3-D solids. We will show that for one special case, finding a pose that satisfies the backward constraints is equivalent to finding a line traversal of the model volumes. This will show that dealing with the backward constraints is at least as difficult as the problem of finding a line traversal of the volumes.

Suppose first that we have an arbitrary set of model volumes, but that every image region consists of only a single point, and that these points are identical, denoted by  $\bar{q}$  (see Fig. 2). The backward constraints are satisfied by a transformation if and only if the 3-D line that projects to  $\bar{q}$  intersects every model volume. Such a transformation exists if and only if there exists a single line that intersects each of the model volumes. This shows that the problem of solving the backward constraints is at least as hard as the problem of determining

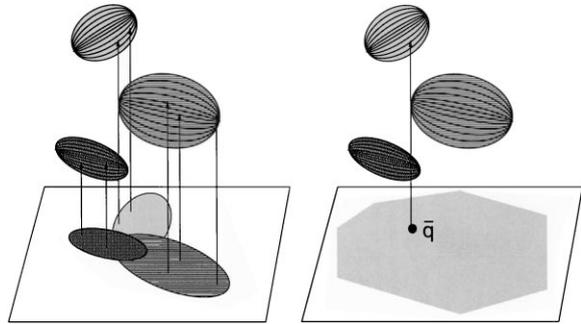


Figure 2. On the left, we show three model volumes projecting to three, overlapping image regions. On the right, we imagine that these regions are occluded, except for one point which they all share. In this case, the problem of solving the backward constraints is equivalent to that of finding a line traversal of the model volumes.

whether a line traversal exists for a set of 3-D volumes. Note that this reasoning applies equally to orthographic or perspective projection.

Unfortunately, Amenta (personal communication) has shown that finding a line traversal to  $n$  arbitrary 3-D volumes is not an LP type problem, because it can contain  $O(n)$  solutions that are disconnected in the space of all possible lines. This implies that gradient descent methods of solving the backward constraints may produce locally optimal solutions that are not globally optimal. Moreover, existing algorithms for finding line traversals are not nearly as efficient as our linear programs, and do not appear to be extendible to the backward constraints without adding considerably greater complexity. For example, Pellegrini (1990) gives an algorithm for finding a line traversal of general polyhedra with edges that have only a constant number of different possible directions. This algorithm requires  $O(n^2 \log(n))$  time for  $n$  polyhedra. It is not clear how to improve upon this result if the polyhedra are restricted to be convex. Solving the forward constraints for such polyhedra would require only  $O(n)$  expected time, because each polyhedron would produce a constant number of constraints, yielding a total of  $O(n)$  constraints, and because fixed dimension linear programs can be solved in  $O(n)$  expected time (Seidel, 1990). And solving the full backward constraints will be much harder than solving the line traversal problem, because the image regions will not, in general, all consist of a single point, and because a full 3-D to 2-D projection must be considered.

Our result is significant because it is not merely a comment on our algorithms; it reflects on the difficulty

of a basic problem in recognition. We have shown that the general problem of determining 3-D model pose in a 2-D image, when one allows for arbitrary occlusions of unknown locus, is at least as hard as a difficult computational geometry problem. Moreover, Amenta's construction (personal communication) uses only volumes that are lines in 3-D. This implies that the problem is difficult even when we restrict ourselves to very simple shapes.

Our result is also somewhat surprising because the forward and backward constraints each encode some of the constraints that together imply a perfect fit between model and image. The backward constraints allow for partial occlusion of the model, while the forward constraints, in some sense allow for partial "occlusion" of the image (some of the image region may go unexplained). Moreover, for the case of planar objects, these constraints are symmetric, and give rise to equally complex problems. It is therefore somewhat unexpected that they would give rise to problems of such different complexity for the case of 3-D objects.

There is, however, a special case for which the line traversal problem can be solved efficiently. This is when the set of 3-D volumes are all axial rectangulars, (i.e., their sides are aligned with the  $x$ -,  $y$ -, and  $z$ -axes). Amenta (1992) gives an efficient algorithm for this case, and subsequently (Megiddo, 1996) has refined this result, showing that this problem can be solved by linear programming. We will now extend this result to propose a useful approximation algorithm for finding object pose, in which we approximate model parts using their bounding axial rectangulars.

Based on Megiddo's method (1996), we can solve the backward constraints using linear programming, for the case where the model volumes are axial rectangulars and the projection model is affine. (Notice that since we are free to change the coordinate frame of the model, in fact we only require the models to be rectangulars with parallel sides.) To do this, it is helpful to think of affine projection in the following way. First, the set of 3-D model points are extended in any one direction to a set of parallel lines; then a 3-D affine transformation is applied to this set of lines so that they are made normal to the image plane. Each 3-D point projects to a 2-D point where its corresponding line intersects the image plane. Now we may think of the inverse of this transformation as applying a 3-D affine transformation to the set of lines normal to the image plane, that intersect each image point. Specifically, let  $\{p_i\}$  denote a set of image points. Let  $\{n_i\}$

denote the corresponding set of lines, so that  $n_i$  is normal to the image plane and includes  $p_i$ . Then  $\{p_i\}$ , are the projection of a set of model points,  $\{q_i\}$ , if and only if there exists a 3-D affine transformation so that each line,  $n_i$ , is transformed to include  $q_i$ . The backward constraints, then, are satisfied by an affine transformation that maps each normal line so that it intersects the corresponding model volume.

We now show how to express the constraint that an image point  $p_i \in R_i$  must be the projection of some point in an axial rectangular,  $V_i$ , as a linear constraint on the inverse of the 3-D affine transformation that maps the model into the scene. We let  $T^{-1}$  denote this inverse transformation, with components denoted by  $t^{-1}$ . Suppose  $V_i$  has a lower corner of  $(x_l, y_l, z_l)$  and an upper corner of  $(x_h, y_h, z_h)$ . Then the transformed line intersects this rectangular if and only if there exists a point on the line whose  $x$ ,  $y$  and  $z$  coordinates lie within these ranges. Consider a line normal to the  $z = 0$  image plane passing through the point  $(u, v)$ . If we transform this line by  $T^{-1}$ , and parameterize it by  $\tau$ , then a point on this transformed line has the coordinates:

$$p_i = (t_{11}^{-1}u + t_{12}^{-1}v + t_x^{-1} + t_{13}^{-1}\tau, \\ t_{21}^{-1}u + t_{22}^{-1}v + t_y^{-1} + t_{23}^{-1}\tau, t_{33}^{-1}\tau),$$

for any choice of  $\tau$ . The backward constraint  $p_i \in TV_i$  is satisfied if and only if it is possible to satisfy the following inequalities:

$$x_l \leq t_{11}^{-1}u + t_{12}^{-1}v + t_x^{-1} + t_{13}^{-1}\tau \leq x_h \\ y_l \leq t_{21}^{-1}u + t_{22}^{-1}v + t_y^{-1} + t_{23}^{-1}\tau \leq y_h \\ z_l \leq t_{33}^{-1}\tau \leq z_h.$$

Assume now that  $t_{13}^{-1}, t_{23}^{-1} > 0$  (in practice we must run four linear programs to account for the different possible signs of  $t_{13}^{-1}$  and  $t_{23}^{-1}$ ). Then we have:

$$\frac{x_l - t_{11}^{-1}u - t_{12}^{-1}v - t_x^{-1}}{t_{13}^{-1}} \leq \tau \leq \frac{x_h - t_{11}^{-1}u - t_{12}^{-1}v - t_x^{-1}}{t_{13}^{-1}} \\ \frac{y_l - t_{21}^{-1}u - t_{22}^{-1}v - t_y^{-1}}{t_{23}^{-1}} \leq \tau \leq \frac{y_h - t_{21}^{-1}u - t_{22}^{-1}v - t_y^{-1}}{t_{23}^{-1}} \\ \frac{z_l}{t_{33}^{-1}} \leq \tau \leq \frac{z_h}{t_{33}^{-1}}.$$

These inequalities are satisfied if and only if every quantity constrained to be less than  $\tau$  is smaller than every quantity constrained to be bigger than  $\tau$ . Namely,

we can replace them with nine inequalities (three of which are trivial) that do not mention  $\tau$  at all. These inequalities appear non-linear, but we can linearize them with a change of variables. Let:

$$\begin{aligned} s_{11} &= \frac{t_{11}^{-1}}{t_{13}^{-1}}, & s_{12} &= \frac{t_{12}^{-1}}{t_{13}^{-1}}, & s_{13} &= \frac{1}{t_{13}^{-1}}, & w_x &= \frac{t_x^{-1}}{t_{13}^{-1}}, \\ s_{21} &= \frac{t_{21}^{-1}}{t_{23}^{-1}}, & s_{22} &= \frac{t_{22}^{-1}}{t_{23}^{-1}}, & s_{23} &= \frac{1}{t_{23}^{-1}}, & w_y &= \frac{t_y^{-1}}{t_{23}^{-1}}, \\ s_{33} &= \frac{1}{t_{33}^{-1}}, \end{aligned}$$

and we get six non-trivial constraints:

$$\begin{aligned} s_{13}x_l - s_{11}u - s_{12}v - w_x &\leq s_{23}y_h - s_{21}u - s_{22}v - w_y \\ s_{13}x_l - s_{11}u - s_{12}v - w_x &\leq s_{33}z_h \\ s_{23}y_l - s_{21}u - s_{22}v - w_y &\leq s_{13}x_h - s_{11}u - s_{12}v - w_x \\ s_{23}y_l - s_{21}u - s_{22}v - w_y &\leq s_{33}z_h \\ s_{33}z_l &\leq s_{13}x_h - s_{11}u - s_{12}v - w_x \\ s_{33}z_l &\leq s_{23}y_h - s_{21}u - s_{22}v - w_y. \end{aligned}$$

These inequalities are homogeneous, reflecting the fact that some components of the transformation are redundant. We can get around this by setting  $t_{33}^{-1} = 1$ , creating difficulties only in a degenerate case. We can therefore express the backward constraints for rectanguloids with parallel sides as a set of linear inequalities, and solve them using linear programming, as we did the forward constraints. For the case of non-rectanguloid model volumes, we can approximate the backward constraints by replacing each model volume with the axial rectanguloid that bounds it.

With this approximation the backward constraints require that the bounding rectanguloid of each model volume project into the image so that it completely contains the corresponding image region. These provide correct constraints on the transformation, since if a volume projects into the image so that it contains a corresponding region, so must its bounding rectangle. The accuracy of the rectangular backward constraints will depend on the specific shape of the model volumes. However, we note that without loss of generality we may apply an arbitrary 3-D affine transformation to the set of model volumes before bounding them, to improve our approximation. In Section 3 we will describe the consequences of this approximation in more detail, showing that in some cases of interest we can find the correct model pose in spite of this approximation.

We are required to run four linear programs, which each correspond to a different visual aspect of the rectanguloids. It is not the case, however, that we can use more complex shapes to approximate the volumes, running a different linear program for each aspect, since the reduction to linear programming depends on decoupling the  $x$  and  $y$  coordinates, not on treating each aspect separately. This is clear also from Amenta's proof (personal communication) that the general problem is not an LP-type problem.

#### 2.4. Local Geometric Features

We now point out that the solution methods described above may be applied to local geometric features as well, provided these can be described as convex sets. There are many existing methods for finding pose using local geometric features. Our solution methods will be suitable only when one wishes to combine information from these features with information based on regions and volumes.

We consider in particular the use of points, corners, and line segments. With the exception of corners, each of these can be thought of as a convex subset of  $\mathcal{R}^3$  that projects to a convex subset of  $\mathcal{R}^2$ . A corner can be thought of as a point with associated vectors, to indicate the directions of the lines forming the corner. These vectors are also convex subsets of the space of vectors.

To apply the forward constraints, one must describe each model feature with one or more points, and each image feature as the intersection of a set of half-planes. Obviously, a model point is trivially described as a point. An image point can be described as the intersection of three half-planes intersecting at the point. A model line segment can be described by its endpoints, which form its convex hull. An image line segment can be described by the intersection of four half-planes. Two half-planes can be along the line segment, but in opposite directions, so that their intersection is the line containing the line segment. Each end point of the line segment can be delimited by another half-plane. Either or both of these can be omitted if it is known that one or both of the line segment's end points is due to an occlusion. The vector giving the orientation of a model corner may also be described using any point along the 3-D ray that describes the corner, which is constrained to lie along a ray in the image, given by the position and direction of the corner. Given such descriptions of local geometric features, the forward constraints are then applied as described above.

Similarly, we may apply the rectangular backward constraints after describing image features using points, and after placing axial rectanguloids about model features. Note that a 3-D point feature can always be exactly described by an axial rectanguloid. A 3-D line segment can be exactly described if it is aligned with one of the axes. Otherwise it may be approximated. The direction vector of a corner can be exactly described in the model if it is aligned with an axis, otherwise an approximation to it will be so large as to be of little value.

### 2.5. *Non-convex Shapes*

For the most part, the solution methods we describe can also be applied to non-convex shapes, if we first approximate them by taking their convex hull. We can apply the forward constraints by requiring that each model volume project inside the convex hull of the corresponding image region. To do this, it is sufficient to require every vertex of the convex hull of the model volume to project inside the half-plane corresponding to every edge bounding the convex hull of the corresponding image region. The rectangular backward constraints can similarly be applied by approximating each model volume using a rectangular box, and using the vertices of the convex hull of the image region.

However, a complication arises in applying the forward constraints, with known occlusion. In that case, we cannot bound the possible extent of a partially occluded image region. If the model volume is non-convex, we cannot assume that by extending the corresponding image region convexly in the direction of an occlusion that we will cover all of the region that has been occluded. It is possible that the occluded portion of the region contains concavities, and extends outside of the largest possible convex region containing the detected region. This is illustrated in Fig. 3. We will not consider this problem further.



*Figure 3.* Suppose the cross-hatched, polygonal area is an image region known to be occluded by the dark rectangle (left). When the corresponding model volume is convex, we may apply the forward constraints, using the largest possible convex region consistent with this occlusion (middle). However, this may be incorrect if the occluded region is not convex (right), and we wish to use the convex hull of the region for the forward constraints.

## 3. Uniqueness of Solutions

We have described two different types of constraints that we may use to efficiently determine model pose. The forward constraints are correct when there is only self-occlusion, or occlusion whose location is known. The backward constraints are correct even in the presence of unknown occlusion. However, these constraints are not complete: the forward constraints do not express the constraints that each model volume should explain every point in each matching image region, while the backward constraints only bound the true set of backward constraints, through the use of axial rectanguloids. We will now none-the-less show that, although incomplete, these constraints are sufficient in many realistic situations to correctly determine model pose.

We first consider the performance of the forward constraints, in the presence of only self-occlusion. We show that, in general, when we have matched three or four volumes, we may determine the correct model pose with linear programming, while, in general, two matches are not sufficient to determine pose uniquely. Next we consider the performance of the backward constraints when the model volumes are each planar (but not mutually coplanar). In this case, some subset of the exact constraints will be present, and we show that in many cases these can determine the correct solution. We also point out that when we apply the rectangular back constraints to curved 3-D volumes they can only produce an approximate solution, and not an exactly correct one. Finally, we point out that the forward constraints with known occlusions can also produce the correct model pose. Throughout our discussion we will mention situations in which degenerate solutions may be found instead of the correct ones. By restricting the solution to represent a rigid transformation we can in many cases use a degenerate solution to determine the correct one. This will be shown in Section 4.

### 3.1. *The Forward Constraints*

In this section, we suppose that  $T$  is a 3-D to 2-D affine transformation that maps the model volumes to the corresponding image regions. This transformation maps curves on the 3-D volumes to the bounding curves of the 2-D regions. These 3-D curves are called the *contour generators*. Our proofs assume the contour generators are closed 1-D curves, which is generically true for convex volumes. With some complication, the proofs may be extended to handle the non-generic case.

We then ask under what circumstances  $T$  is the unique transformation that satisfies the forward constraints. When  $T$  is unique, this means that using a linear program to find a transformation satisfying the forward constraints must in fact produce  $T$ , the correct solution. We had previously (Basri and Jacobs, 1997) reported preliminary results on this problem for the special case of planar model volumes that are not mutually coplanar. We now extend those results to the case of fully three-dimensional model volumes.

**Theorem 1.** *Suppose the transformation  $T$  maps the four model volumes,  $V_0, V_1, V_2, V_3$  to the four image regions,  $R_0, R_1, R_2, R_3$ , and there does not exist a plane that intersects all four volumes. Then,  $T$  is the only transformation that satisfies the forward constraints.*

This is proven in Appendix A.

**Theorem 2.**  *$T$  is not the only transformation satisfying the forward constraints if and only if, for the contour generators  $T$  produces, there exists a plane,  $P$ , such that, for all contour generators, either  $P$  contains the contour generator, or  $P$  intersects it in two points, such that the orthographic projection onto  $P$  of the tangents to the contour generators at the intersection points are all parallel. This implies that for three model volumes viewed in general position,  $T$  will be the unique transformation that satisfies the forward constraints.*

This is proven in Appendix A.

We now consider the case in which only two model volumes are matched to two image regions. We do not provide necessary and sufficient conditions for these to determine a unique transformation. However, we do note that any two model volumes may be viewed so that their corresponding image regions intersect. When this occurs, the forward constraints will be satisfied by any transformation that shrinks the model volumes to a very small size, and projects them inside this region of intersection. Therefore, any two model volumes will lead to a non-unique solution when viewed from a significant range of viewpoints.

### 3.2. The Rectangular Backward Constraints

We now address the problem of determining when the rectangular backward constraints are sufficient to correctly determine the pose of a model. Recall that the

backward constraints bound each model volume with an axial rectangular, and then require a valid transformation to map this rectangular into the image so that it completely contains the corresponding image region.

Let  $C_i$  denote the axial rectangular that bounds the volume  $V_i$ . Note that the contour generator for  $C_i$  will consist of six line segments, or four line segments for those special viewpoints in which only one face of the rectangular is visible. This means that the projection of  $C_i$  will be a six (or four) sided convex polygon. The rectangular backward constraints require that  $R_i \subseteq TC_i$ . Since  $V_i \subseteq C_i$ , the correct transformation will satisfy these constraints.

In general,  $TV_i$  may lie completely inside  $TC_i$ . In fact,  $TV_i$  and  $TC_i$  will touch only in the case where  $V_i$  touches one of the line segments that form the boundary of  $C_i$  (or for special viewpoints in which a side of  $C_i$  projects to a line segment in the image). For a general smooth 3-D volume, no bounding rectangular (or polygon of any sort) will touch the volume in one of its edges, since this would require a discontinuity in the volume. On the other hand, a convex planar volume can always be oriented so that a bounding rectangular is also planar, and touches it in at least four points. Therefore, cases of interest exist in which either  $TV_i$  is always completely inside  $TC_i$  or in which  $TV_i \cap TC_i$  contains at least four points.

Suppose first that  $R_i = TV_i$  lies entirely inside  $TC_i$ , for all volumes  $V_i$ . In this case, it is obvious that any small perturbation to the transformation  $T$  will not violate the rectangular backward constraints. Therefore, in such a situation the backward constraints will not uniquely determine the correct pose, although they may still produce a good approximation to this pose.

This is not surprising; after all the rectangular backward constraints involve an approximation, and so one expects them to produce poses with some error. Surprisingly, though, we can also show that for an interesting class of volumes the rectangular backward constraints will produce exactly the correct answer. Suppose that the volumes are all planar, and that they each lie in either of two different planes. This is a common occurrence if the volumes are either surface markings on two different faces of a polyhedron, or on the walls or ceiling of a room. We show that such volumes can lead to a unique solution to the backward constraints.

In particular, we show that when two or more model volumes are planar, and lie in the same plane, there can be a unique transformation of that plane that satisfies the rectangular backward constraints. If in addition there is another planar model volume, lying in

a different plane, there can generically be a unique solution to the rectangular backward constraints. A precise statement and proof of this can be found in Appendix B. In brief, when planar volumes lie in two different planes we can, without loss of generality, assume that these planes are axial. In this case, axial rectangles touch the boundary of the model volumes at points that always project to the boundary of image regions. Therefore, the rectangular backward constraints include some tight constraints that can allow only a unique transformation to be valid.

### 3.3. *The Forward Constraints, with Known Occlusion*

We now point out that our discussion of the backward constraints also provides an example of a situation in which the forward constraints, with known occlusion, can lead to a unique transformation. When we use the forward constraints with known occlusions we are making use of a subset of the constraints that would be available in the absence of occlusion. As we have described, the rectangular backward constraints can include a small subset of the complete backward constraints. For planar volumes, those points that are extremal in the axial directions lead to tight constraints on the model pose. Suppose now that the same extremal points that we make use of in Section 3.2 are visible in the image. In this case, exactly the same reasoning applies to show that a unique transformation will satisfy the forward constraints. Consequently, if any but these extremal points are occluded, we may still apply the forward constraints to find the correct transformation. This provides an example that shows that even with a large amount of occlusion, the forward constraints can produce the correct transformation, provided that this occlusion is identified.

In general the forward constraints with known occlusions will be much more effective in determining a unique solution than will the rectangular backward constraints, since every unoccluded point on the boundary of the image regions will provide a tight constraint on the transformation. It is, however, beyond the scope of this paper to fully characterize when these constraints lead to a correct solution.

## 4. Recovering from Degeneracies

We now consider how we may handle certain situations in which the model volumes lead to a non-unique

solution. The most common cause of this occurs when the model itself is in a sense degenerate, and so a 3-D affine transformation that satisfies analogs to the forward and backward constraints may be found. However, we will show that in this case we can often reconstruct the correct transformation, undoing the effects of this degeneracy.

Suppose there exists a 3-D to 3-D affine transformation,  $A \neq I$ , such that  $AV_i \subseteq V_i$  for all  $i$ . In this case, the forward constraints can almost never lead to a unique solution. If the correct transformation is  $T$ , then the transformation  $TA$  will clearly also satisfy the forward constraints. Moreover,  $TA \neq T$  except for special cases when projection completely removes the effects of  $A$ .

If the model contains three distinct volumes, any such transformation must contain three fixed points. If the volumes are not traversed by a single line,  $A$  must consist of a fixed plane, with a contraction in some direction toward that plane.

Similarly, a degeneracy will almost always occur in the rectangular backward constraints when there is an affine transformation  $A$  such that  $V_i \subseteq AC_i$ . This typically occurs when a plane exists that intersects the  $C_i$  rectangularoids on sides that can be divided into two parallel sets. In this case  $A$  can be the affine transformation that leaves this plane fixed, and expands the model in the direction shared by the rectangularoid sides intersected by the plane. For example, suppose all the rectangularoids have minimum  $x$  coordinates less than zero, and maximum  $x$  coordinates greater than zero. In this case, the  $x = 0$  plane will intersect each  $C_i$  in two sides parallel to the  $y = 0$  plane and in two sides parallel to the  $z = 0$  plane. In this case, we may expand the model rectangularoids in the  $x$  direction (which is the intersection of the  $y = 0$  plane and the  $z = 0$  plane) so as to leave the  $x = 0$  plane fixed. This transformation has the form:

$$A\vec{p} = \begin{pmatrix} 1 + \epsilon & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \vec{p}$$

for any  $\epsilon > 0$ . In this example,  $V_i \subseteq C_i \subseteq AC_i$ . In such a situation, whenever  $T$  satisfies the rectangular backward constraints, so will  $TA$ .

However, since the model is accessible to us in advance, we may detect when such degeneracies occur, and undo their effect. To determine the possibility of such a degeneracy, we may apply the forward or backward constraints for the case of 3-D affine

transformations. For the forward constraints we compare the model to itself. For the backward constraints we compare the model volumes to their bounding rectangularoids. Since 3-D affine transformations are linear, these constraints can be solved for using linear programming, and will reveal the presence of a transformation such as  $A$ . However, in the experiments described below, we have simply detected this possibility by hand.

If a model can be contracted or expanded perpendicularly in a single direction, we first preprocess the model so that this direction is aligned with one of the axes. Without loss of generality, assume this is the  $x$ -axis. Let the contraction/expansion transformation applied to the model be given by

$$A\vec{p} + \vec{b} = \begin{pmatrix} 1 + \epsilon & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \vec{p} + \begin{pmatrix} a_x \\ 0 \\ 0 \end{pmatrix}.$$

Notice that we must allow for an unknown translation of  $a_x$  in the  $x$  direction, as a part of the contracting/expanding transformation. This is because, although we assume that contraction or expansion is in the  $x$  direction, and that there is a fixed plane perpendicular to the direction, we do not assume that this is the  $x = 0$  plane. In some cases there will not be a unique plane that might be the fixed plane, there might be a family of parallel planes any one of which might have been fixed in the contraction/expansion of the model. Now, suppose the image is obtained by applying a rigid transformation to the model followed by a scaled orthographic projection. The imaging process can be written as

$$S\vec{p}' + \vec{t} = \begin{pmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \end{pmatrix} \vec{p}' + \begin{pmatrix} t_x \\ t_y \end{pmatrix},$$

where the entries  $s_{ij}$  are the first two rows of a scaled rotation matrix. Our solution methods, then, will produce a transformation  $T$  that is a composition of the two transformations, as follows:

$$\begin{aligned} T\vec{p} &= S(A\vec{p} + \vec{b}) + \vec{t} \\ &= \begin{pmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \end{pmatrix} \left( \begin{pmatrix} 1 + \epsilon & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \vec{p} + \begin{pmatrix} a_x \\ 0 \\ 0 \end{pmatrix} \right) \\ &\quad + \begin{pmatrix} t_x \\ t_y \end{pmatrix}. \end{aligned}$$

Given the transformation,  $T$ , we wish to recover the matrix  $S$  that indicates the true scaled orthographic projection of the model. This is easily done, since

$$\begin{aligned} s_{12} &= t_{12} & s_{22} &= t_{22} \\ s_{13} &= t_{13} & s_{23} &= t_{23}. \end{aligned}$$

We may then readily determine the values of  $s_{11}, s_{21}$  which will satisfy the rigidity constraints of the matrix, so that:

$$\begin{aligned} s_{11}s_{21} + s_{12}s_{22} + s_{13}s_{23} &= 0 \\ s_{11}^2 + s_{12}^2 + s_{13}^2 &= s_{21}^2 + s_{22}^2 + s_{23}^2 \end{aligned}$$

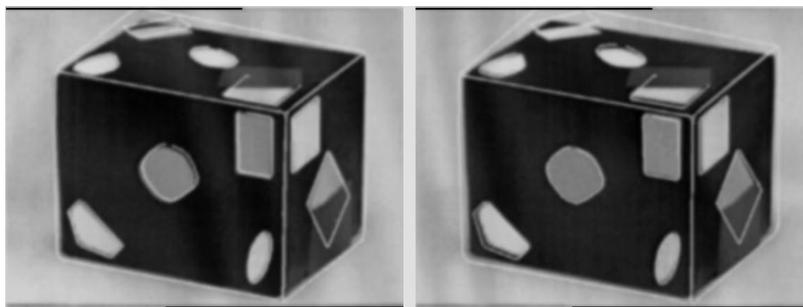
Thus we can recover the scaled rotation matrix and also the  $y$  translation that produced the image regions. We cannot directly recover the translation  $a_x$ . However, once we have determined the rest of the transformation, it is easy to determine the appropriate translation. For example, we could run linear programming again, allowing for only an  $x$  translation.

## 5. Experiments

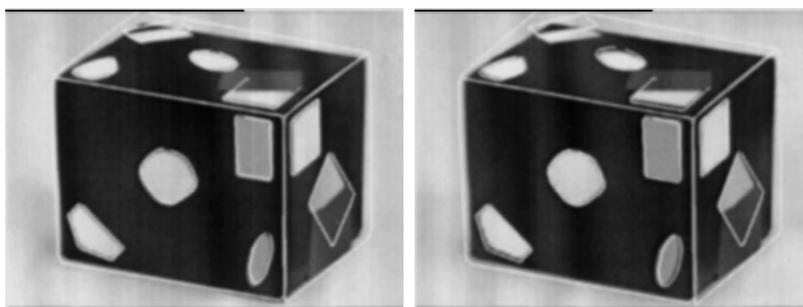
We now present some experiments that demonstrate the feasibility of our approach to recognition. These experiments will provide useful information about the accuracy of the poses that we can recover using both the forward and the rectangular backward constraints.

In these experiments a model was first constructed by hand, using images of the object and knowledge of its structure. Then, the Canny edge detector (Canny, 1986) was run on a new image of the object. We automatically extracted sets of edges that formed salient convex groups, using the grouping system described in (Jacobs, 1996). The localization of groups in the image will therefore contain errors due to running a real edge detector and grouping system on real images. A subset of these groups were then extracted and matched by hand to groups in the model. See Nayar and Bolle (1996) for one suggestion about how to use intensity information to match such regions automatically. Our system then used these matches to determine the model poses shown here.

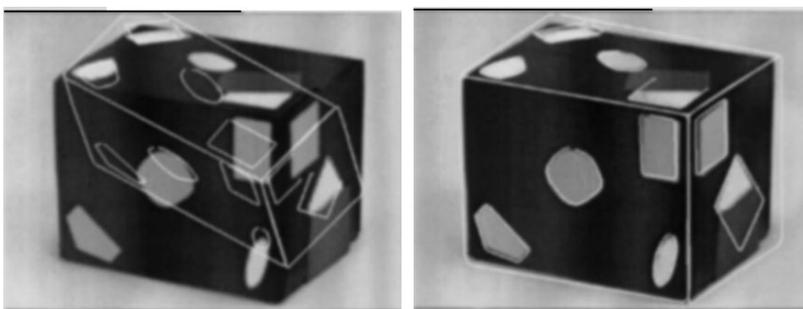
In the first set of experiments we use a black box with different shaped regions painted on the side. Although this object is somewhat artificial, it allows us to experiment with a variety of different conditions. In Figs. 4–10 we show the poses derived by the forward and backward constraints using different combinations



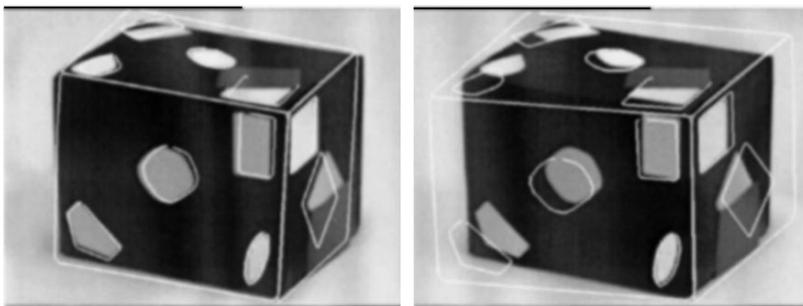
*Figure 4.* This shows poses derived using the forward and backward constraints. The image is shown, with the regions used to derive the pose marked with a white hatching. Superimposed over the image are the white outlines of all model groups, to indicate the pose that has been derived using the forward constraints (left) and backward constraints (right). This figure shows the poses derived using five regions.



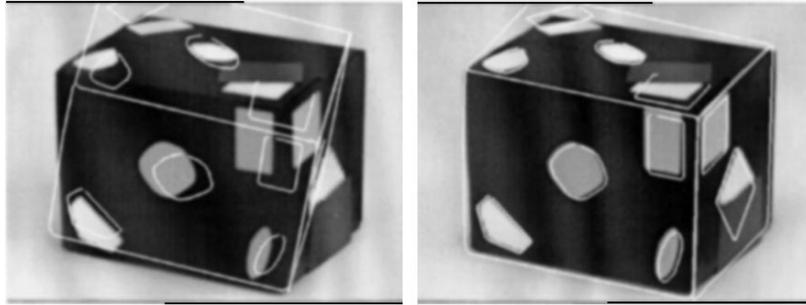
*Figure 5.* Another example using five regions. The pose derived using the forward constraints is on the left, and the pose from the backward constraints is on the right.



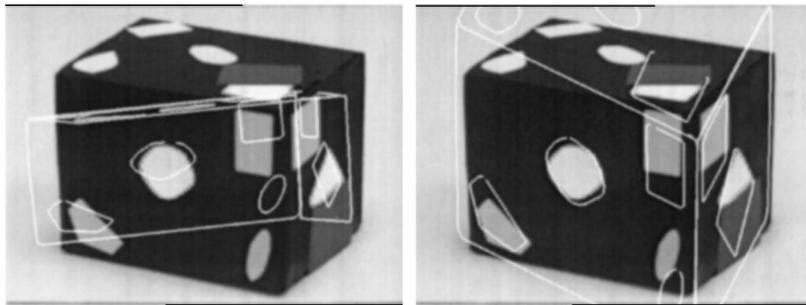
*Figure 6.* An example using four regions. Note that one of the regions is partially occluded, leading to a poor solution with the forward constraints (left).



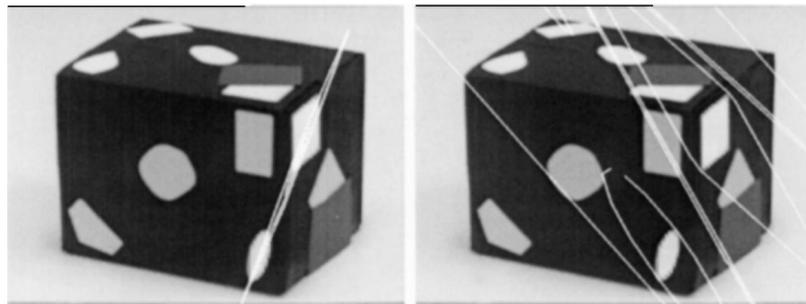
*Figure 7.* An example using three regions. The backward constraints (right), which are approximate, lead to a much noisier solution than the forward (left).



*Figure 8.* An example using three regions, one of them occluded. This leads to a poor solution for the forward constraints (left), which do not allow for occlusion. The backward constraints (right), produce a much more accurate pose. Note that in spite of the occlusion, the backward constraints produce a more accurate solution than they did in Fig. 7 because a more stable triple of image regions are matched.



*Figure 9.* An example using three regions, with two of them occluded. This leads to a very poor solution for the forward constraints (left), while the solution for the backward constraints also degrades. Occlusion may cause the correct solution to become non-unique, and indeed we can see that the pose found satisfies the backward constraints very well.



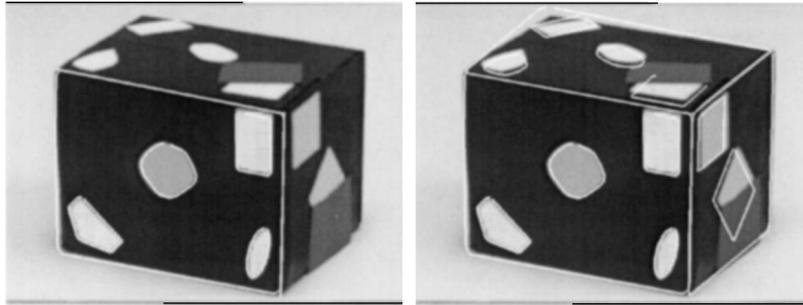
*Figure 10.* This example shows poses derived from two regions. We can see that these are not sufficient to determine the correct pose.

of regions. This shows that both sets of constraints are able to produce accurate model poses when at least three region correspondences are used. It also illustrates how the poses degrade as we use fewer regions, and as the amount of occlusion increases. Of course the forward constraints are especially vulnerable to unidentified occlusions.

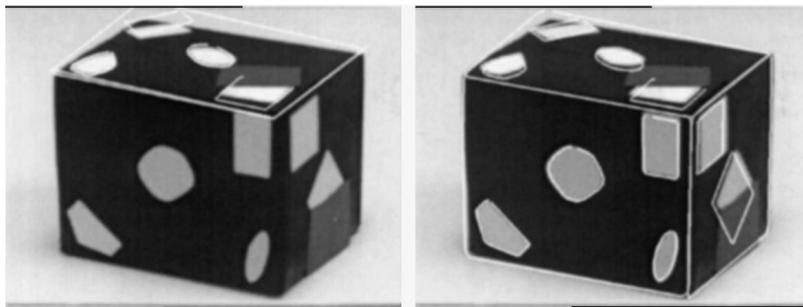
In Figs. 11–13 we demonstrate the method discussed in Section 4 for determining the correct scaled

orthographic projection from a degenerate transformation. The model regions used in these examples are coplanar, and thus lead to a degeneracy. Note, however, that degeneracies can occur even when 3-D model volumes are used.

Figure 14 shows the forward and backward constraints being used to recognize two other objects. These experiments show that even with noise due to the edge detection and grouping process, we can use



*Figure 11.* Here we show a set of regions that produce a degenerate solution. The forward constraints are applied. However, since the model regions matched are really coplanar, a solution is found that contracts the 3-D model into a single plane (left). Since this potential degeneracy can be detected ahead of time, we may postprocess the pose to “uncontract” it, producing a scaled orthographic transformation (right) that matches the model and image better.



*Figure 12.* Similarly, we show the backward constraints applied to produce a degenerate solution (left), along with the “uncontracted” pose.



*Figure 13.* Similar results for a different object. The pose of the soda can found by the forward constraints (left) is significantly contracted. After “uncontracting” the pose (right) into a scaled orthographic projection, we obtain a better fit.

region matches to accurately determine pose without explicit correspondences between local features such as points or line segments.

Finally, Figs. 15–18 show the results of our algorithms for a synthetic model composed of 3-D volumes. The object is composed of seven volumes: the four legs, the body, the neck and the head. That is, each volume is a part; the faces of volumes are not used as separate parts. We segment and match these parts

perfectly in the images. Being synthetic, this object is simple to model accurately. But moreover, since the only source of error is digitization in the projection, any inaccuracies that we find are due to limitations of the methods proposed. Figure 16 shows that the backward constraints can produce very accurate poses, in spite of the approximations made by taking bounding rectangularoids. Note that none of the model parts are at all rectangular; the legs and neck, for example are

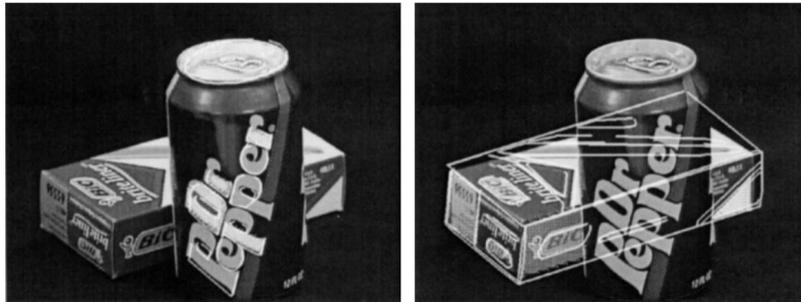


Figure 14. These figures show the performance of the system on more realistic objects. On the left, the system uses the forward constraints to accurately determine the pose of the soda can. Four regions are used in this case. The regions are surface markings on the cylinder of the can, and a circular region from the top of the can. On the right, the backward constraints are used to locate the pen box. Note that the soda can is occluding some of the regions used.

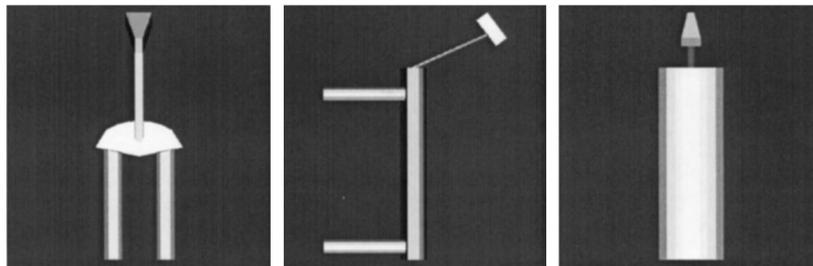


Figure 15. A simple synthetic animal, seen from along the  $x$ -axis (left)  $y$ -axis (center) and  $z$ -axis (right). Bounding boxes for the rectangular backward constraints were built with the model in this reference frame.

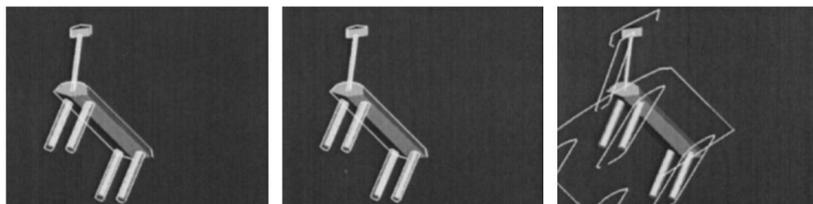


Figure 16. These figures show the rectangular backward constraints applied to a synthetic, 3-D object. We show the image used with some of the boundary of the volumes of the projected model superimposed in white. On the left, the pose found using all seven model volumes. In the center, we use four volumes: the head, the two left legs, and the front right leg. In both cases, we find accurate poses. On the right, we use only the head and the two left legs, and a poorer pose is found.

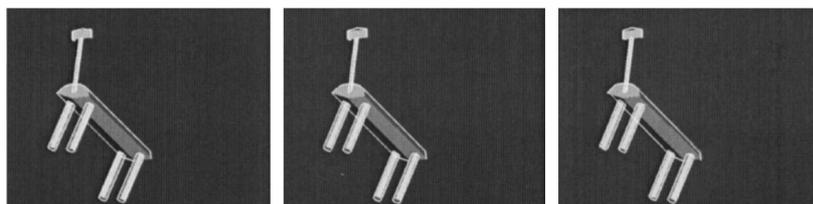
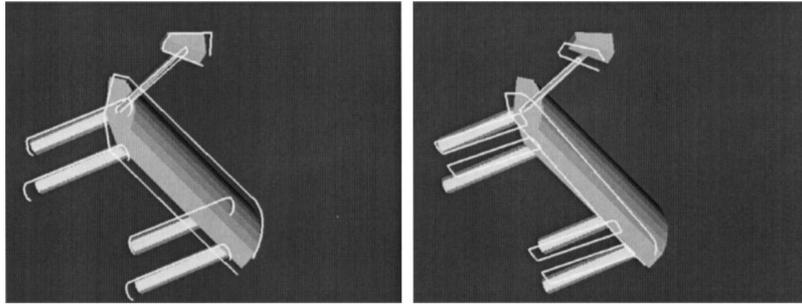


Figure 17. The results using the forward constraints using the same image and volume/region matches as in the previous figure. Since there is little occlusion of objects parts by other parts, the forward constraints produce accurate poses. When only three volumes (right) are used, they continue to produce an accurate pose. Note that the backward constraints lead to a much less accurate pose in this case, as shown in the previous figure.



*Figure 18.* Here we show a view of the object in which the leg volumes are significantly occluded by the body. We match all seven volumes to regions. The rectangular backward constraints (left) produce a fairly accurate pose. The forward constraints (right), which do not allow for such occlusion, produce a much noisier pose.

nine and seven sided polygons, respectively. Figure 17 shows that the forward constraints do indeed produce accurate results when only three volumes are matched. Finally, Fig. 18 shows the performance of the methods when some model volumes are partially occluded, in this case by other parts of the object.

## 6. Conclusion

Recognition of 3-D objects in 2-D images has been hampered by the difficulty of finding representations that can faithfully model complex 3-D objects and still be used to determine pose based on their 2-D images. In this work we make use of a simple representation, which divides objects into parts and then represents each part as a volume of points. This representation can clearly be applied to a large class of objects. Our contribution is to show that it can also be used to accurately determine the pose of these objects. We show that even without specific correspondences between local geometric features, we may use region matches to determine the correct model pose. At the same time our method allows us to incorporate correspondences between points, lines or line segments, should they be available.

Specifically, we present new results that show that the forward constraints, which allow for self-occlusion, may correctly determine the pose of a 3-D object, typically when a correspondence has been found between three 3-D parts of the object and three matching 2-D image regions. We have also shown that it is a more difficult problem to find poses that satisfy the backward constraints, which allow for unidentified occlusions. This problem can have multiple disconnected solutions. These results apply not just to our algorithm

but to any parts-based recognition system which determines pose while allowing for arbitrary image occlusions of unknown locus. However, we have also devised a novel algorithm, based on work in computational geometry, that finds solutions that satisfy an approximation to the backward constraints, using linear programming. And we show that in some cases of interest, this approximate solution leads to our finding exactly the correct model pose. These results demonstrate that we can recognize 3-D objects using a very simple, and novel representation of their structure.

## Appendix A: Uniqueness of Forward Constraints

In this section, we prove Theorems 1 and 2. We first prove the following useful lemma:

**Lemma 1.** *Suppose that  $T'$  is a transformation that also satisfies the forward constraints. Then for each model volume  $V_i$  and image region  $R_i$ , there exists a point,  $\vec{p}_i \in V_i$  such that  $T\vec{p}_i = T'\vec{p}_i$ .*

**Proof:** Throughout this section, we will assume that the *contour generator* of  $V_i$  is a 1-D curve on  $V_i$  that projects to the 1-D boundary of  $R_i$ . This will be true for smooth, generic, convex volumes in general position. Our reasoning can be readily extended to other cases, however we will not consider those in this paper to simplify our arguments. Clearly we can construct a 2-D surface, call it  $S_i$ , such that  $S_i$  is bounded by the contour generator, such that  $S_i \subseteq V_i$  and such that  $TS_i = R_i$ . Therefore,  $T$  will define a continuous one-to-one mapping between  $S_i$  and  $R_i$ . So even though  $T$  is not in general invertible, we may invert  $T$  when we restrict its domain to  $S_i$ . Let  $\bar{T}$  denote the restriction of

$T$  to this domain, and let  $\bar{T}^{-1}$  denote its inverse, a mapping from  $R_i$  to  $S_i$ . Because  $T'$  satisfies the forward constraints,  $T'S_i \subseteq R_i$ . Therefore  $T'\bar{T}^{-1}$  defines a continuous mapping from  $R_i$  into  $R_i$ . Brouwer's fixed point theorem, a basic result in functional analysis (see, for example, Conway (1990)), tells us that since  $R_i$  is convex, and hence topologically equivalent to a disc, such a mapping must have a fixed point. That is, there exists some point  $\vec{q}_i \in R_i$  such that  $\vec{q}_i = T'\bar{T}^{-1}\vec{q}_i$ . Therefore,  $T(\bar{T}^{-1}\vec{q}_i) = T'(\bar{T}^{-1}\vec{q}_i)$ , proving the lemma.  $\square$

Using this lemma, we may now show the following:

**Theorem 1.** *Suppose the transformation  $T$  maps the four model volumes,  $V_0, V_1, V_2, V_3$  to the four image regions,  $R_0, R_1, R_2, R_3$ , and there does not exist a plane that intersects all four volumes. Then,  $T$  is the only transformation that satisfies the forward constraints.*

**Proof:** Let  $T'$  map the four model volumes  $V_0, \dots, V_3$  to inside the corresponding image regions  $R_0, \dots, R_3$ . By Lemma 1 for every volume  $V_i$ ,  $i = 0, \dots, 3$  there exists a point  $\vec{p}_i \in V_i$  such that  $T'(\vec{p}_i) = T(\vec{p}_i)$ . Since there exists no plane that intersects all four volumes the points  $\vec{p}_0, \dots, \vec{p}_3$  are not coplanar. Consequently, since correspondences of four non-coplanar points determine a 3-D to 2-D affine transformation uniquely then  $T = T'$ .  $\square$

We now show that typically, only three matches are required to uniquely determine the pose. In (Basri and Jacobs, 1995) we have proven the following lemma:

**Lemma 2.**  *$T$  is uniquely determined for the set of volumes  $V_i$  if and only if it is uniquely determined for the set of volumes  $QV_i$ , where  $Q$  is any 3-D affine transformation.*

This tells us that we may, without loss of generality, place the model volumes in any affine reference frame. This can significantly simplify our reasoning, since it allows us to assume without loss of generality that  $T$  is the identity transformation, plus orthographic projection. We now suppose that we have matched three model volumes to three image regions, and that the image regions are not all intersected by a single line. Further, we suppose that the image regions were produced by applying the transformation,  $T$ , to the model volumes, and that a different transformation,  $T' \neq T$ ,

also satisfies the forward constraints. Then there exist three non-collinear model points,  $\vec{p}_0, \vec{p}_1, \vec{p}_2$  such that  $T\vec{p}_0 = T'\vec{p}_0$ ,  $T\vec{p}_1 = T'\vec{p}_1$ , and  $T\vec{p}_2 = T'\vec{p}_2$ . Lemma 2 tells us that we may assume, without loss of generality, that these three points are in the  $z = 0$  plane and fixed under the transformation  $T$ , and therefore also under  $T'$ . This means that the entire  $z = 0$  plane is fixed under these two transformations. Further, without loss of generality, we may assume that  $T(0, 0, 1) = (0, 0)$ . Therefore, we may write:

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

and

$$T' = \begin{pmatrix} 1 & 0 & k_1 \\ 0 & 1 & k_2 \end{pmatrix},$$

where either  $k_1, k_2$  or both are non-zero.

By choosing this affine reference frame, we have constructed things so that each contour generator,  $s_i$ , projects orthographically to form the boundary of the corresponding region. Let  $\vec{p}$  be an arbitrary point on  $V_0$ 's contour generator, with coordinates  $(x, y, z)$ . Then  $T\vec{p} = (x, y)$ , and  $T'\vec{p} = (x, y) + z(k_1, k_2)$ . This tells us that  $T'$  maps the contour generator so that it is displaced from the region boundary in either the direction  $(k_1, k_2)$  or  $-(k_1, k_2)$ , depending on whether the point is above or below the image plane.

We may use this fact to place constraints on the direction of the contour generator's tangent. If one of the contour generators is either entirely above, or entirely below the image plane, then clearly  $T'$  will map some of these points outside the corresponding region, violating our assumptions.

Next, suppose that the image plane intersects the contour generator in at least two discrete points, but not in an entire subcurve of the contour generator. Let  $R_i \cap s_i$  be the points  $\vec{p}_i^1, \vec{p}_i^2$ . Let the tangent to  $R_i$  at  $\vec{p}_i^j$  be  $\vec{w} = (w_x, w_y, 0)$ . Let the tangent to  $s_i$  at the point  $\vec{p}_i^j$  have the direction  $\vec{v}$ . Then the directions of  $\vec{w}, T\vec{v}$  and  $T'\vec{v}$  must all be the same. That is true because if a point in the model projects to the boundary of the image region, and the tangents of the region point and the projected model point differ, then the projected model volume will not be contained in the image region. So, since  $\vec{w} = T\vec{v}$ , we must have  $\vec{v} = (w_x, w_y, v_z)$  for some  $v_z$ . The points  $\vec{p}_i^j$  are also fixed under  $T'$ , since they lie in the  $z = 0$  plane which is fixed under  $T'$ . Therefore, the tangent to  $T'V_i$  at  $\vec{p}_i^j$

is  $(w_x + k_1 v_z, w_y + k_2 v_z)$ . Again, the condition that  $T'V_i \subseteq R_i$  implies that  $T'\vec{v}$  must have the same direction as  $\vec{w}$ . Since either  $k_1 \neq 0$  or  $k_2 \neq 0$  it follows that the directions of  $(w_x, w_y)$  and  $(k_1, k_2)$  must be parallel. Therefore, the tangents to each region  $R_i$  at a point  $\vec{p}_i^j$  must all be parallel to  $(k_1, k_2)$ , and so they must all be parallel to each other.

If the image plane intersects the contour generator in an entire curve, we may apply the same reasoning to the two end points of the curve. In the special case that the entire contour generator lies in the image plane, then that volume will not constrain  $k_1$  and  $k_2$ , for  $T'$  close to  $T$ .

This has shown:

**Theorem 2.** *T is not the only transformation satisfying the forward constraints if and only if, for the contour generators T produces, there exists a plane, P, such that, for all contour generators, either P contains the contour generator, or P intersects it in two points, such that the orthographic projection onto P of the tangents to the contour generators at the intersection points are all parallel.*

Simple variable counting now tells us that for general shapes, in general position, this will not be possible. Given a set of model volumes and a particular viewpoint, the contour generators will be fixed. First, for general objects the contour generator will not be planar. Next, we have three degrees of freedom in choosing a plane to intersect the contour generators. Each plane will determine the direction of six tangent vectors projected into that plane. For all these tangents to be parallel provides five degrees of constraint; however we have only three degrees of freedom available to satisfy these constraints. Therefore, in general, given a set of model volumes and a view of them, there will be a unique transformation that satisfies the forward constraints, provided that the image regions cannot be intersected by a single line.

**Appendix B: Uniqueness of Backward Constraints**

We begin our analysis by supposing that at least two of the model volumes,  $V_0, V_1$  are planar and lie in the same plane. Without loss of generality, we assume that this is the  $z = 0$  plane. Note that we are free to preprocess the model volumes with an affine transformation

to achieve this before approximating them with rectangles. The volumes will then be approximated by 2-D rectangles in this plane, which we call  $C_0$  and  $C_1$ . Each side of each rectangle will touch the corresponding volume in at least one point. These will be the points with the highest and lowest values in their  $x$  and  $y$  coordinates. We will call these points  $\vec{x}_{0,l}, \vec{x}_{0,h}, \vec{y}_{0,l}, \vec{y}_{0,h}, \vec{x}_{1,l}, \vec{x}_{1,h}, \vec{y}_{1,l}, \vec{y}_{1,h}$ , where, for example,  $\vec{x}_{0,h}$  is the point in  $V_0$  with the highest  $x$  coordinate (see Fig. 19). Note that if there is a line segment on the boundary of one of the volumes with constant  $x$  or  $y$  values we can pick any of these points, or we can easily extend the reasoning given below to include this case. Also, one point may be extremal in both the  $x$  and  $y$  direction; this does not effect the argument given below. At these eight points, the backward constraints will require that the corresponding region point found in the image will lie on the appropriate side of the projection of a horizontal or vertical line passing through that point in the model. Therefore, we will have eight constraints that are not approximate, but that are subsets of the constraints we would have if we

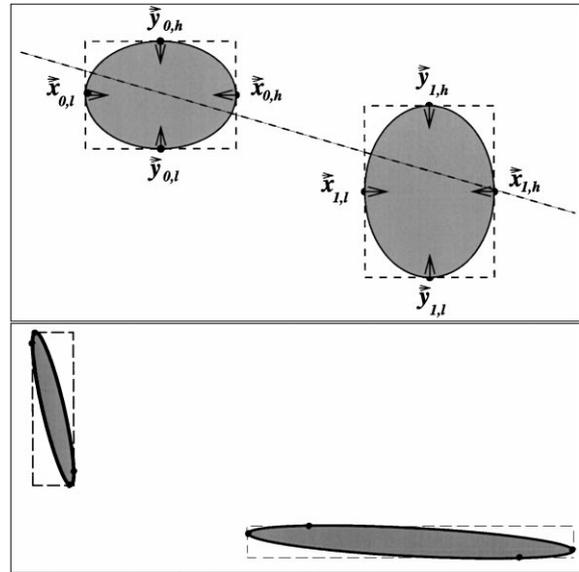


Figure 19. Two coplanar volumes  $V_0$  and  $V_1$  bounded with flat rectanguloids  $R_0$  and  $R_1$ . The points  $\vec{x}_{0,l}, \vec{x}_{0,h}, \vec{y}_{0,l}, \vec{y}_{0,h}, \vec{x}_{1,l}, \vec{x}_{1,h}, \vec{y}_{1,l}, \vec{y}_{1,h}$  are the contact points of the volumes and the rectanguloids. In the case on the top, the two pairs of points  $\vec{y}_{0,l}, \vec{y}_{1,l}$  and  $\vec{y}_{0,h}, \vec{y}_{1,h}$  are linearly separable, as is indicated by the dashed line. On the bottom, the volumes lead to a unique solution in the  $z = 0$  plane.

could apply the backward constraints exactly. A transformation infinitesimally different from the correct one will not violate any of the other rectangular backward constraints, but it may violate these ones.

Now we consider when these eight constraints will suffice to uniquely determine that portion of the projection that effects the  $z = 0$  plane. From Lemma 2 we may assume, without loss of generality, that  $T$  leaves the  $z = 0$  plane fixed;<sup>1</sup> we must then determine the circumstance under which there exists a different transformation,  $T'$ , which has a different effect on the  $z = 0$  plane while respecting the rectangular backward constraints. It will be convenient to focus on the inverse of  $T'$  when we restrict its effect to the  $z = 0$  plane. We ask when there is an inverse transformation that can map each image point to the appropriate side of the bounding model rectangle. We denote this inverse transformation  $T'^{-1}$ , with components  $t'^{-1}$  given the appropriate subscript.

Note that the  $x$  coordinates of points in the  $z = 0$  plane are effected only by  $t'_{11}{}^{-1}$ ,  $t'_{12}{}^{-1}$  and  $t'_x{}^{-1}$ , while the  $y$  coordinates are changed by  $t'_{21}{}^{-1}$ ,  $t'_{22}{}^{-1}$  and  $t'_y{}^{-1}$ . Since the  $\vec{x}_{0,h}$ ,  $\vec{x}_{0,l}$ ,  $\vec{x}_{1,h}$ ,  $\vec{x}_{1,l}$  points are constrained only in the  $x$  direction, and the corresponding  $y$  points are constrained only in the  $y$  direction, we may consider these two sets of points separately. A non-unique solution exists if and only if either there exist values of  $t'_{11}{}^{-1}$ ,  $t'_{12}{}^{-1}$  and  $t'_x{}^{-1}$  that map the points  $\vec{x}_{0,h}$ ,  $\vec{x}_{0,l}$ ,  $\vec{x}_{1,h}$ ,  $\vec{x}_{1,l}$  within the minimum and maximum  $x$  values of their bounding rectanguloids, or if there similarly exist values of  $t'_{21}{}^{-1}$ ,  $t'_{22}{}^{-1}$  and  $t'_y{}^{-1}$  that map the  $y$  points in the appropriate  $y$  directions.

The transformation  $T'^{-1}$  will change the  $x$  coordinate of the point  $(x, y)$  by the amount:

$$T'^{-1}(x, y, 0) - x = (t'_{11}{}^{-1} - 1)x + t'_{12}{}^{-1}y + t'_x{}^{-1}.$$

Therefore, the line

$$(t'_{11}{}^{-1} - 1)x + t'_{12}{}^{-1}y + t'_x{}^{-1} = 0$$

will have its  $x$  coordinates fixed (note that in the case of pure translation we may think of the equation  $t'_x{}^{-1} = 0$  as describing a vertical line at  $x = \infty$ ). On one side of this line, the value of  $(t'_{11}{}^{-1} - 1)x + t'_{12}{}^{-1}y + t'_x{}^{-1}$  is positive, on the opposite side this value is negative. This means that either all points will shift their  $x$  coordinate towards this fixed line, or all points will shift away from it. Therefore, the only way such a transformation can

satisfy the backward constraints is if we can draw a line that separates the points  $\vec{x}_{0,l}$ ,  $\vec{x}_{1,l}$  from the points  $\vec{x}_{0,h}$ ,  $\vec{x}_{1,h}$ . In that case any such separating line can be the fixed line, and  $T'^{-1}$  can map all points towards that line, in the  $x$  direction. It may or may not be possible to find such a separating line, depending on the configuration of the model.

Exactly the same reasoning holds for the points  $\vec{y}_{0,l}$ ,  $\vec{y}_{0,h}$ ,  $\vec{y}_{1,l}$ ,  $\vec{y}_{1,h}$ . Therefore, there exist models for which the rectangular backward constraints uniquely determine the transformation of the  $z = 0$  plane, as illustrated in Fig. 19. While it is somewhat difficult to find such a configuration in two model volumes, similar reasoning holds when more than two volumes are coplanar, except that now a larger set of points must be linearly separable for a transformation that satisfies the backward constraints to be non-unique.

Suppose now that the model contains a set of volumes for which the backward constraints uniquely determine the transformation of the  $z = 0$  plane. This tells us that any transformation,  $T' \neq T$  satisfying the backward constraints must have the form:

$$T' \vec{p} = \begin{pmatrix} 1 & 0 & t'_{13} \\ 0 & 1 & t'_{23} \end{pmatrix} \vec{p},$$

with either  $t'_{13} \neq 0$  or  $t'_{23} \neq 0$ . We now ask when an additional planar model volume, not in the  $z = 0$  plane, will suffice to completely determine the model-to-image transformation. Without loss of generality we may assume that this volume lies in the  $y$ - $z$  plane, so that it is bounded by a rectangle in this plane, and has four extremal points in the  $y$  and  $z$  directions. Furthermore, we may assume that the transformation  $T$  has the form:

$$T \vec{p} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \vec{p}.$$

That is,  $T$  is the identity transformation when applied to the  $x$ - $y$  plane, while it maps the  $y$ - $z$  plane to the  $x$ - $y$  image plane by mapping  $(0, y, z)$  to  $(z, y)$ . Therefore,  $T$  acts also like an identity transformation on the  $y$ - $z$  plane, while converting it into the  $x$ - $y$  plane. Now consider the effect that  $T'$  has as it maps the  $y$ - $z$  plane into the  $x$ - $y$  plane. It leaves the  $z = 0$  line fixed, so  $T'$  can consist only of a contraction of the  $y$ - $z$  plane towards this line, along with an interchange of coordinates. If both extremal points of the model volume in the  $z$  direction are on the same side of the line  $z = 0$ , then

any contraction or expansion towards or away from that line will violate one of the constraints imposed by the bounding rectanguloid. That is, once the backward constraints imply a unique solution within one plane, they fail to produce a unique solution only when this plane intersects the remaining volumes. This will never happen if the volumes are surface markings on a convex polyhedra, as for example, when they lie on the walls or ceiling of a room.

We have now shown that, though approximate, the rectangular backward constraints may still uniquely determine the correct model to image transformation in situations of real interest. At the same time, unlike our previous 2-D formulation of the backward constraints, we can now integrate information from planar volumes that are not coplanar, or from non-planar volumes.

### Acknowledgment

The authors wish to thank Baskaran Vijayakumar for his assistance in building the experimental system and in running the experiments. This research was supported by the Unites States-Israel Binational Science Foundation, Grant No. 94-100. The vision group at the Weizmann Inst. is supported in part by the Israeli Ministry of Science, Grant No. 8504. Ronen Basri is an incumbent of Arye Dissentshik Career Development Chair at the Weizmann Institute.

### Note

1. This point is somewhat subtle. We may assume this because the bounding rectanguloids can be axial to any affine reference frame, which need not be the natural Euclidean reference frame in which we store the model.

### References

- Alter, T.D. and Jacobs, D. 1998. Uncertainty propagation in model-based recognition. *International Journal of Computer Vision*, 27(2):127–159.
- Amenta, N. 1992. Finding a line traversal of axial objects in three dimensions. In *Proc. of the Third ACM-SIAM Symp. on Discrete Alg.*, pp. 66–71.
- Amenta, N. Personal communication.
- Amenta, N. 1994. Bounded boxes, Hausdorff distance, and a new proof of an interesting Helly-type theorem. In *Proceedings of the 10th Annual ACM Symposium on Computational Geometry*, pp. 340–347.
- Basri, R. 1996. Paraperspective  $\equiv$  affine. *International Journal of Computer Vision*, 19(2):169–179.
- Basri, R. and Jacobs, D.W. 1995. Recognition using region correspondences. Technical Report CS95-33, The Weizmann Institute of Science.
- Basri, R. and Jacobs, D. 1996. Matching convex polygons and polyhedra, allowing for occlusion. *First ACM Workshop on Applied Computational Geometry*, pp. 57–66.
- Basri, R. and Jacobs, D. 1997. Recognition using region correspondences. *International Journal of Computer Vision*, 25(2):145–166.
- Bergevin, R. and Levine, M. 1993. Generic object recognition: Building and matching coarse descriptions from line drawings. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(1):19–36.
- Biederman, I. 1985. Human image understanding: Recent research and a theory. *Computer Graphics, Vision, and Image Processing*, (32):29–73.
- Binford, T. 1971. Visual perception by computer. In *IEEE Conf. on Systems and Control*.
- Brooks, R. 1981. Symbolic reasoning among 3-D models and 2-D images. *Artificial Intelligence*, 17:285–348.
- Canny, J. 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698.
- Clemens, D. 1991. Region-based feature interpretation for recognizing 3D models in 2D images. *MIT AI TR-1307*.
- Conway, J.B. 1990. *A Course in Functional Analysis*. Springer-Verlag.
- Duda, R.O. and Hart, P.E. 1973. *Pattern Classification and Scene Analysis*. Wiley-Interscience Publication, John Wiley and Sons, Inc.
- Dudani, S.A., Breeding, K.J., and McGhee, R.B. 1977. Aircraft identification by moments invariants. *IEEE Transactions on Computations*, C-26(1):39–46.
- Fischler, M.A. and Bolles, R.C. 1981. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Com. of the A.C.M.*, 24(6):381–395.
- Forsyth, D. 1996. Recognizing algebraic surfaces from their outlines. *International Journal of Computer Vision*, 18(1):21–40.
- Forsyth, D., Mundy, J., Zisserman, A., and Rothwell, C. 1992. Recognising rotationally symmetric surfaces from their outlines. *European Conf. on Comp. Vis.*, pp. 639–647.
- Gross, A. and Boulton, T. 1990. Recovery of generalized cylinders from a single intensity image. In *Proc. of the DARPA IU Workshop*, pp. 557–564.
- Horand, R. 1987. New methods for matching 3-D objects with single perspective views. *IEEE Trans. Pattern Anal. Machine Intell.*, 9(3):401–412.
- Hu, M.K. 1962. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8:169–187.
- Huttenlocher, D., Klanderman, G., and Rucklidge, W. 1993a. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863.
- Huttenlocher, D., Noh, J., and Rucklidge, W. 1993b. Tracking non-rigid objects in complex scenes. In *4th Int. Conf. on Computer Vision*, pp. 93–101.
- Huttenlocher, D.P. and Ullman, S. 1990. Recognizing solid objects by alignment with an image. *Int. J. Computer Vision*, 5(2):195–212.
- Jacobs, D. 1992. Recognizing 3-D objects using 2-D images. *MIT AI Memo 1416*.

- Jacobs, D. 1996. Robust and efficient detection of salient convex groups. *IEEE Trans. Pattern Anal. Machine Intell.*, 18(1):23–37.
- Jacobs, D. 1997. Matching 3-D models to 2-D images. *Int. J. Computer Vision*, 21(1/2):123–153.
- Koenderink, J. and van Doorn, A. 1991. Affine structure from motion. *Journal of the Optical Society of America*, 8(2):377–385.
- Kriegman, D. and Ponce, J. 1990. On recognizing and positioning curved 3-D objects from image contours. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(12):1127–1137.
- Lamdan, Y. and Wolfson, H.J. 1988. Geometric hashing: A general and efficient model-based recognition scheme. In *Second International Conference Computer Vision*, pp. 238–249.
- Lowe, D. 1985. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers: The Netherlands.
- Marr, D. and Nishihara, H. 1978. Representation and recognition of the spatial organization of three dimensional structure. In *Proceedings of the Royal Society of London B*, 200:269–294.
- Megiddo, N. 1996. Finding a line of sight thru boxes in d-space in linear time. Research Report RJ 10018, IBM Almaden Research Center, San Jose, California.
- Nagao, K. and Grimson, W. 1994. Object recognition by alignment using invariant projections of planar surfaces. In *12th Int. Conf. on Pattern Rec.*, pp. 861–864.
- Nayar, S. and Bolle, R. 1996. Reflectance based object recognition. *Int. J. of Comp. Vis.*, 17(3):219–240.
- Pellegrini, M. 1990. Stabbing and ray shooting in three dimensional space. In *Proc. of the 6th Annual Symp. on Comp. Geometry*, pp. 177–186.
- Pentland, A. 1987. Recognition by parts. In *Proceedings of the First International Conference on Computer Vision*, pp. 612–620.
- Persoon, E. and Fu, K.S. 1977. Shape discrimination using Fourier descriptors. *IEEE Transactions on Systems, Man and Cybernetics*, 7:534–541.
- Poelman, C.J. and Kanade, T. 1997. A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):206–218.
- Ponce, J., Chelberg, D., and Mann, W. 1989. Invariant properties of straight homogeneous generalized cylinders and their contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(9):951–966.
- Reeves, A.P., Prokop, R.J., Andrews, S.E., and Kuhl, F.P. 1984. Three-dimensional shape analysis using moments and fourier descriptors. In *Proc. of Int. Conf. on Pattern Recognition*, pp. 447–450.
- Richard, C.W. and Hemami, H. 1974. Identification of three dimensional objects using fourier descriptors of the boundry curve. *IEEE Transactions on Systems, Man and Cybernetics*, 4(4):371–378.
- Rivlin, E., Dickinson, S.J., and Rosenfeld, A. 1995. Recognition by functional parts. *Computer Vision: Image Understanding*, 62(2):164–176.
- Rock, I. 1983. *The Logic of Perception*. MIT Press: Cambridge, MA.
- Rothwell, C., Forsyth, D., Zisserman, A., and Mundy, J. 1993. Extracting projective structure from single perspective views of 3D point sets. In *Third Int. Conf. on Comp. Vis.*, pp. 573–582.
- Rucklidge, W. 1997. Efficiently locating objects using the Hausdorff distance. *Int. J. of Comp. Vis.*, 24(3):251–270.
- Sadjadi, F.A. and Hall, E.L. 1980. Three-dimensional moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(2):127–136.
- Seidel, R. 1990. Linear programming and convex hulls made easy. In *Proc. of the Sixth Annual Symp. on Comp. Geometry*, pp. 211–215.
- Shafer, S. and Kanade, T. 1983. The theory of stright homogeneous generalized cylinders. Tech. Report CS-083-105, Carnegie Mellon Univ.
- Solina, F. and Bajcsy, R. 1990. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Trans. on PAMI*, 12(2):131–146.
- Sugimoto, A. 1996. Object recognition by combining paraperspective images. *Int. J. of Comp. Vis.*, 19(2):181–201.
- Terzopoulos, D. and Metaxas, D. 1991. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Trans. on PAMI*, 13(7):703–714.
- Thompson, D. and Mundy, J. 1987. Three-dimensional model matching from an unconstrained viewpoint. In *Proceedings IEEE Conference Rob. Aut.*, pp. 208–220.
- Tomasi, C. and Kanade, T. 1992. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154.
- Ulipinar, F. and Nevatia, R. 1995. Shape from contour: Straight generalized cylinders and constant cross section generalized cylinders. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):120–135.
- Ullman, S. and Basri, R. 1991. Recognition by linear combinations of models. *IEEE Trans. on PAMI*, 13(10):992–1006.
- Vijayakumar, B., Kriegman, D., and Ponce, J. 1995. Invariant-based recognition of complex curved 3D objects from image contours. In *Fifth Int. Conf. on Comp. Vis.*, pp. 508–514.
- Zerroug, M. and Nevatia, R. 1994. Using invariance and quasi-invariance for the segmentation and recovery of curved objects. In *Applications of Invariance in Computer Vision*, J. Mundy and Z. Zisserman (Eds.), Springer-Verlag: Berlin, Heidelberg.