3D Geometric Constraints from Surface Patch Matching

Robert B. Fisher

Dept. of Artificial Intelligence, University of Edinburgh

5 Forrest Hill, Edinburgh EH1 2QL, Scotland, United Kingdom

November 29, 1990

Abstract

Some three dimensional scene analysis programs pair data surface patches (e.g. as extracted from range data) to model surface patch features. From the correspondences, a variety of rotation and translation constraints on the model instance can be deduced, depending on the assumptions behind the correspondences. This paper elaborates some of the constraints and their associated assumptions for planar, cylindrical, conical and generic surface patches. For example, if one knows that a planar model patch must overlap a planar data patch, then this constrains the alignment of the model surface normal and the possible translations of the model patch. The constraints are expressed algebraically for patches that are: nearby, touch, contained, or registered. The constraints are used to estimate object position, given a single patch correspondence. Examples of their use and evaluation are given.

Keywords: three dimensional scene analysis, position estimation, geometric reasoning

1 Introduction

Fisher and Orr [12] have been investigating the use of algebraic inequalities as a representation of partial position constraints, much as Brooks did in his ACRONYM system [4].

This research is being undertaken as part of the IMAGINE II scene analysis system ([10], [11]), which is currently concerned with surface-based object recognition.

The data primitives in the matching are surface patches, which can be obtained by a variety of techniques for segmenting range data (e.g. [3], [5], [17], [19]). These processes result in data patches with consistent curvature signs. Objects are defined using the SMS models [9], which allow hierarchical object representations, with surface patches being one of the primitive features.

The model matching process is not described here, but makes correspondences between single (and multiple) model and data surface patches. With these correspondences, it is possible to estimate the position of the object containing the patches (assuming the correct model has been chosen and the correspondences were made correctly) or reject the correspondences as inconsistent.

A key element of the geometric analysis is formulating algebraic constraints between a rotated and translated model and the observed surface patches. The constraints attempt to define or refine the set of allowable positions for the object consistent with the observed data patches. For example, pairing a planar data patch with a planar model patch immediately constrains the rotations allowable, because the transformed model normal $\vec{n}_m$ must be aligned with the data normal $\vec{n}_d$, within an angular observation error ϵ. Hence,

$$T\vec{n}_m \cdot \vec{n}_d \geq cos(\epsilon)$$

is a constraint on the rotation component of the position T (here expressed as a prefix transformation on the model normal $\vec{n}_m$.) We assume throughout this paper that all surface patches are oriented (i.e. they have an outside face with which the normal is associated).

There are a variety of constraints on both the rotation and translation, depending on the assumptions made about the relationship between the model and data patches, such as whether the patches overlap or one is contained in the other. The generation of the constraints is the responsibility of the matching program, which then depends on the constraint analysis program for solving them. Each model-to-data correspondence may introduce several constraints; moreover, there may be several correspondences that each add their own constraints. Altogether, an analysis may generate many constraints which then have to be solved to generate a single position estimate.

As there are a variety of model and data surface patches, there are a variety of constraints that can be generated. This paper presents and discusses the algebraic position constraints that we have identified and are using, under a variety of assumptions about the correspondence. In particular, we present:

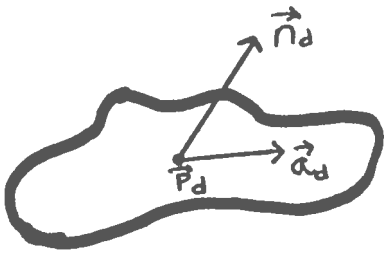
- the descriptions of the model and data surface patches,
- the geometric constraints deducible from the model-to-data patch correspondences,
- a summary of one approach to solving the constraint based on a value-propagating network acting on interval [1] values,
- some experiments showing the performance of the constraints and
- a few examples of applying the theory and solving the resulting constraints in the geometric reasoning networks.

These are described individually in sections 2 through 7.

It should be noted that the types of constraints described here differ from those described by Grimson and Lozano-Perez [15], who identified a set of geometric constraints that **pruned the search space of model-to-data correspondences**, whereas here we give constraints that **limit the spatial location and orientation of the object.** Their constraints are also useful, and we use them in the model matching process that leads to the generation of the constraints identified below.

2 Data Surface Patch Types And Descriptions

In this paper, we assume that the features extracted from the image data are surface patches, characterized by their curvature and extent. These patches will be suitable for direct correspondence with the SMS model features presented in the next section. A summary of how we extract the surface patches is given in Section 7. However, in brief, they are found using a curvature-sign based local classification of smoothed (discontinuity segmented) range data, with region growing to produce larger patches.



Central point (center of mass):	$\vec{p}_d$	distance error (isotropic):	$\vec{\gamma}$
Normal at $\vec{p}_d$:	$\vec{n}_d$	angular error:	ϵ
Arbitrary axis:	$\vec{a}_d$	angular error:	β
Oriented bounding width:	$\omega_d(\phi)$	distance error:	δ_ω

Figure 1: Planar Patch Descriptions

There are four types of data surface features considered here: planar, cylindrical, conical and generic patches. We now describe the features and how they are characterized.

The planar patch (see Figure 1) is flat and has bounded extent. By considering the patch boundary, it is possible that a data patch might only be paired with a model patch in one way, and we would like to exploit as much information as possible. However, we would also like to use a simple scheme that can be used with all patches, irrespective of shape. Hence, we represent planar patches by the bounding width ($\omega_d(\phi)$) at each orientation (ϕ) (see Figure 2) for our position constraints, which can lose some of the information in irregular patches. This width is a measure of the maximum extent of the surface in each direction. The orientation $\phi = 0$ is chosen arbitrarily, and each width measure has an error range of $\pm\delta_w$ associated with it. Additionally, we use the position ($\vec{p}_d$) of the center-of-mass (with the isotropic error range $\pm\vec{\gamma}$), the direction $\vec{n}_d$ of the surface normal at that point (with angular error $\pm\epsilon$) and an arbitrary direction $\vec{a}_d$ in the patch (observed with angular error $\pm\beta$).

The cylindrical patch (see Figure 3) is a patch on the surface of a cylinder and has bounded extent. We simplify the patch by bending a rectangular bounding patch onto the infinite cylinder containing the actual patch, where the sides of the bounding patch are

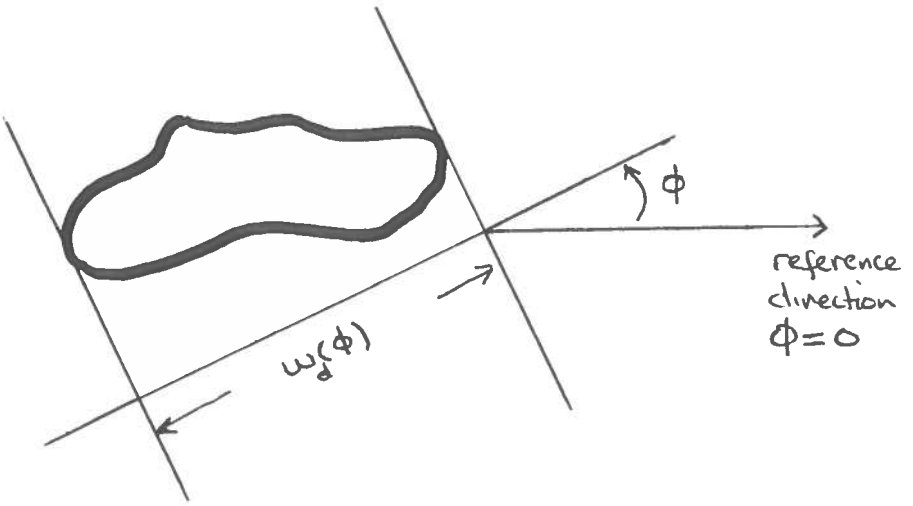


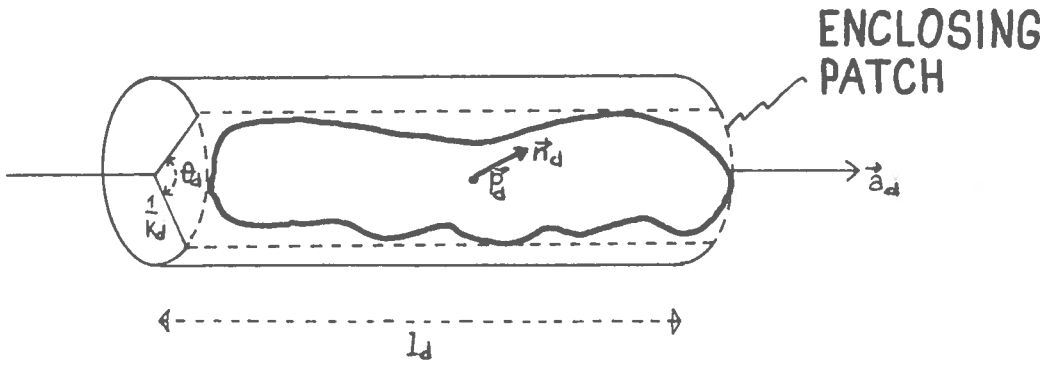
Figure 2: Width Bounding of An Irregular Figure at One Orientation

either parallel to the axis of the cylinder or are parts of circular cross-sections through the cylinder. The measurements taken from the patch are the length of the patch along the axis (l_d with error $\pm\delta_l$), the angular width of the patch across the axis (θ_d with error $\pm\sigma$) and the cylinder's curvature (κ_d). No error is associated with the curvature, because the constraints assume that the correct patch correspondences have been made, and hence the model curvature (κ_m) can be used.

Additionally, we use the position of the central point ($\vec{p}_d$ with error $\pm\vec{\gamma}$), the direction of the surface normal at that point ($\vec{n}_d$ with error $\pm\epsilon$) and the direction of the cylinder's axis ($\vec{a}_d$ with error $\pm\beta$). Here, the central point is defined by the mean value of the radial coordinate expression for the surface.

If only a fragment of a cylinder patch is observed or the observed patch is part of a noise-corrupted patch with a different shape, then it may be misclassified. Hence, we also include the bounding dimensions of the patch when flattened onto a plane ($\omega_d(\phi)$ with error $\pm\delta_\omega$) and arbitrary axis ($\vec{a}_{pd}$ with error $\pm\beta_\omega$). The projection plane is the one tangential to the surface at $\vec{p}_d$.

The conical patch (see Figure 4) is a patch on the surface of a cone and has bounded extent. We simplify the patch by bending a wedge-shaped bounding patch onto the infinite cone containing the actual patch, where the sides of the bounding patch are either rulings down the sides of the cone or are meridians around the cone. We use the patch center



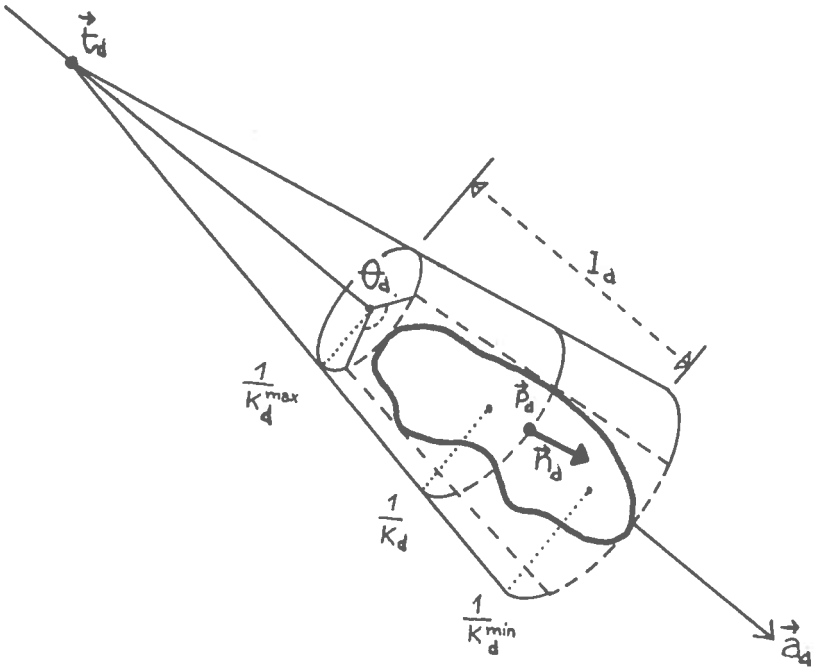
Central point (center of mass):	$\vec{p}_d$	distance error (isotropic):	$\vec{\gamma}$
Normal at $\vec{p}_d$:	$\vec{n}_d$	angular error:	ϵ
Curvature axis:	$\vec{a}_d$	angular error:	β
Curvature:	κ_d		
Enclosing cylinder length:	l_d	distance error:	δ_l
Enclosing cylinder angular width:	θ_d	angular error:	σ
Enclosing flat patch width:	$\omega_d(\phi)$	distance error:	δ_ω
Flattened patch axis:	$\vec{a}_{pd}$	angular error:	β_ω

Figure 3: Cylindrical Patch Descriptions

position ($\vec{p}_d$ with error $\pm\vec{\gamma}$) the direction of the surface normal at point $\vec{p}_d$ ($\vec{n}_d$ with error $\pm\epsilon$) and the direction of the cone's axis ($\vec{a}_d$ with error $\pm\beta$). The patch center is defined as the point in the patch that has the mean value (over the patch) of $(h\rho\theta, h)$, where h is the distance from the cone tip of the projection onto the cone axis and θ is the angular position of the point around the axis.

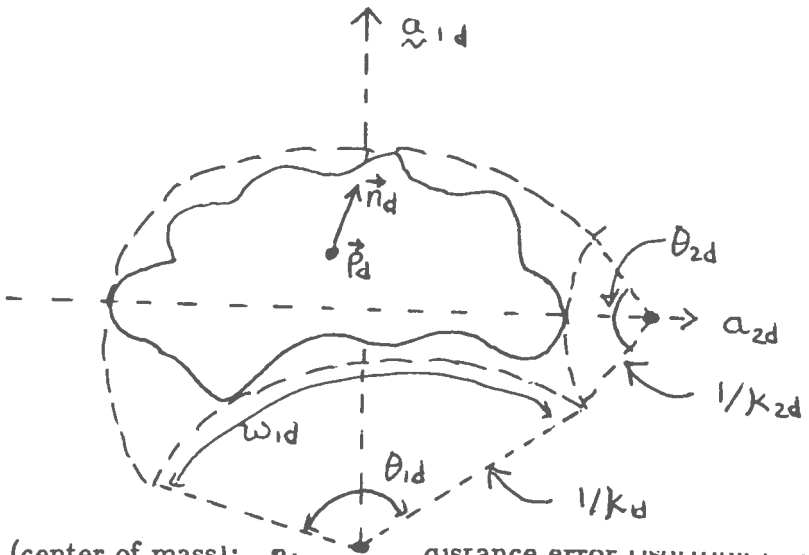
The measurements taken from the patch are the length of the patch along the axis (l_d with error $\pm\delta_l$), the maximum angular width of the patch across the axis (θ_d with error $\pm\sigma$), the cone's curvature at the point ($\vec{p}_d$) (κ_d with error $\pm\alpha$) and the radius rate of change (ρ_d) (i.e. $radius = \rho_d * distance\_from\_tip$). No error is associated with the radius rate, because the constraints assume that the correct patch correspondences have been made, and hence the model radius rate (ρ_m) can be used.

The cone model primitive includes the position of the tip of the cone ($\vec{t}_d$), if visible. Also, the minimum (κ_{mind}) and maximum (κ_{maxd}) curvatures of the cone are known. Similarly to the cylindrical patch, we also take bounding dimensions of the flattened conical patch ($\omega_d(\phi)$ with error $\pm\delta_\omega$) with axis ($\vec{a}_{pd}$ with error $\pm\beta_\omega$).



Cone tip:	$\vec{t}_d$		
Central point (center of mass):	$\vec{p}_d$	distance error (isotropic):	$\vec{\gamma}$
Normal at $\vec{p}_d$:	$\vec{n}_d$	angular error:	ϵ
Cone axis:	$\vec{a}_d$	angular error:	β
Curvature at $\vec{p}_d$:	κ_d	distance error:	α
Radius rate:	ρ_d		
Enclosing cone length:	l_d	distance error:	δ_l
Enclosing cone angular width:	θ_d	angular error:	σ
Maximum Curvature:	κ_{maxd}		
Minimum Curvature:	κ_{mind}		
Enclosing flat patch width:	$\omega_d(\phi)$	distance error:	δ_ω
Flattened patch axis:	$\vec{a}_{pd}$	angular error:	β_ω

Figure 4: Conical Patch Descriptions



Central point (center of mass):	p_d	distance error (isotropic):	γ
Normal at $\vec{p}_d$:	$\vec{n}_d$	angular error:	ϵ
Curvature axes:	$\vec{a}_{1d}, \vec{a}_{2d}$	angular errors:	β_1, β_2
Curvatures at $\vec{p}_d$:	κ_{1d}, κ_{2d}		
Enclosing patch widths:	w_{1d}, w_{2d}	distance errors:	δ_1, δ_2
Enclosing patch angular widths:	θ_{1d}, θ_{2d}	angular errors:	σ_1, σ_2
Enclosing flat patch width:	$\omega_d(\phi)$	distance error:	δ_ω
Flattened patch axis:	$\vec{a}_{pd}$	angular error:	β_ω

Figure 5: Generic Curved Patch Descriptions

The generic patch (see Figure 5) is a patch on the surface of a generic surface curved in two directions and has bounded extent. We simplify the patch by bending a 'rectangular' bounding patch onto the infinite surface containing the actual patch, where the sides of the bounding patch are formed by intersecting planes perpendicular to the curvature directions (at the nominal central point). We use a patch center ($\vec{p}_d$ with error $\pm\vec{\gamma}$), the direction of the surface normal at that point ($\vec{n}_d$ with error $\pm\epsilon$) and the directions of curvature at the central point ($\vec{a}_{1d}$) and ($\vec{a}_{2d}$ with errors $\pm\beta_1$ and $\pm\beta_2$). The measurements taken from the patch are the linear widths (w_{1d}, w_{2d} with errors $\pm\delta_1$ and $\pm\delta_2$) and angular widths (θ_{1d}, θ_{2d} with errors $\pm\sigma_1$ and $\pm\sigma_2$) of the patch about the central point, and the curvatures (κ_{1d}, κ_{2d}) at the point. Here, the generic patch has no perfect mapping to a flattened bounding patch; however, since the flattened patch is used only when the curvatures are close to zero, we will use the patch defined by the orthographic projection onto the plane passing through the center of mass with normal $\vec{n}_d$. This patch has cross-section widths $\omega_d(\phi)$ with error $\pm\delta_\omega$ and axis $\vec{a}_{pd}$ with error $\pm\beta_\omega$.

3 Model Surface Patch Types And Descriptions

The constraints given in Section 4 are designed for use with the surface primitives of the SMS object representation system [9] which form the basis for the **IMAGINE II** system [10] currently being developed. We briefly describe SMS and then define the information used in this paper.

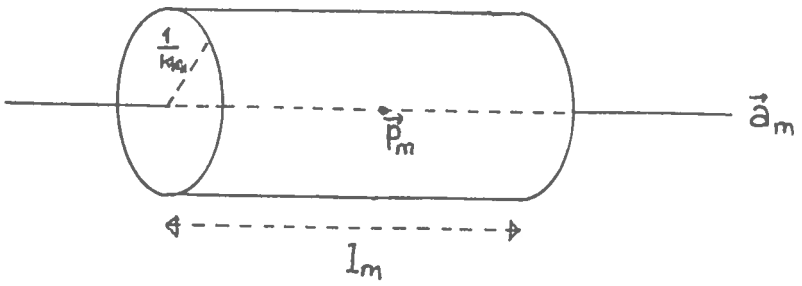
SMS is a geometric modeling system, based on multiple, alternative representations (curve, surface or first and second-order volumetric primitives). The models need not have all features specified, and the features are usually represented in a subcomponent hierarchy. The features may have parameterized sizes and properties (e.g. surface curvatures) and there may be partially or fully unconstrained degrees-of-freedom in the reference frame transformations. The models also record (1) multiple levels of scale-based representation in a simplification hierarchy and (2) viewpoint dependent visibility information.

The goal of SMS is to represent the visual aspects of an object that characterize its identity, rather than describe its shape. Hence, the modeling approach aims to provide representations that closely correspond to reliably extractable image features.

Following ACRONYM [4], all numerical values can be symbolic variables or expressions, as well as constants. This has been used for a class of generic model representation, by allowing size variation amongst the recognizable objects. Variables are defined as either local or global in a model and are bound by a dynamic scoping mechanism. Each SMS model can include a set of constraints on the variables, which are expressed in algebraic form.

SMS provides several types of model surface primitives, of which five types are considered here: planar, cylindrical, full cylinder, cone and generic patches. One use of the surface primitives is to provide model features that can be directly matched to data surface patches, which is the use assumed in this paper. The intent is to provide model features that closely correspond to extractable data features to reduce the representation transformation 'distance'.

The model surface patches are characterized similarly to the data patches (as described in Section 2) except that they are represented with a m subscript instead of a d subscript. Not all of the shape information is currently modeled explicitly in the SMS scheme, but



Central point (center of mass):	$\vec{p}_m$
Curvature axis:	$\vec{a}_m$
Curvature:	κ_m
Enclosing cylinder length:	l_m
Enclosing flat patch width:	$\omega_m(\phi)$
Flattened patch axis:	$\vec{a_{pm}}$

Figure 6: Cylinder Descriptions

all are straightforward and trivial extensions.

In addition to the four data patch primitive shapes, a full cylinder primitive exists, which represents cylindrical surfaces that extend completely around the object, and hence has no orientation about its axis. The description of the cylinder (see Figure 6) is similar to the cylindrical patch. We simplify by bounding the actual feature with a section of cylinder cut perpendicularly to the axis. The measurements taken from the cylinder are the length (l_m) of the patch and the cylinder's curvature (κ_m). Additionally, we use the position of the patch center ($\vec{p}_m$), and the direction of the cylinder's axis ($\vec{a}_m$). Here, the patch centre is defined to be the intersection of the axis with the plane perpendicular to the axis that divides the mass of the cylinder in half. We also include the bounding dimensions of the hemi-cylinder when flattened onto a plane ($\omega_m(\phi)$) with axis ($\vec{a_{pm}}$).

4 Constraints From The Different Patch Pairings

For us, the key data type in geometric reasoning for high-level computer vision is the position. It represents the relative spatial relationship between two visual features (e.g.

world-to-camera, camera-to-model, or model-to-subcomponent). Positions are represented using a 3-vector representing for relative translation and a unit quaternion (4-vector) for relative orientation (of the form $(cos(\theta/2), sin(\theta/2)\vec{w})$ for a rotation of θ about the axis $\vec{w}$).

The key geometric relationships concern relative position and have two forms – *exact* and *partially constrained.* An example of an exact form is: let object A be at global position $(\vec{r}_A, \vec{t}_A)$, (rotation $\vec{r}_A$ and translation $\vec{t}_A$) and object B be at $(\vec{r}_{AB}, \vec{t}_{AB})$ relative to A. Then, the global position of B is:

$$(\vec{r}_B, \vec{t}_B) = (r_{AB} * r_A, r_{AB} * t_A * r'_{AB} + t_{AB})$$

where $*$ is the quaternion multiplication operator and " $'$ " is the quaternion inverse operator.

A partially constrained position is given by an inequality constraint, such as:

$$t_{Az} \geq 50$$

This means that the z component of A's global position is at least 50. Such a constraint might arise from some *a priori* scene knowledge, or from observation of a fragment of a surface.

Other relationships concern vectors or points linked by a common transformation, as in $T\vec{v}_1 = \vec{v}_2$, or the proximity of points or vectors:

$$|\vec{p}_1 - \vec{p}_2| < \epsilon$$

Complications arise because each data measurement may have some error or uncertainty, and hence the estimated values may also have these. Alternately, a variable may be only partially constrained in the model or by *a priori* scene information. Hence, each numerical quantity is represented by an interval [1] of possible values.

In our current implementation of the IMAGINE II system (a design sketch is given in [10]), we represent geometric position quantities as intervals because the interval makes explicit the uncertainity in the value. Constraints are represented as inequalities over the intervals, much as in the ACRONYM system [4]. The benefits of the interval and algebraic approach include: (1) a uniform representation for point and vector quantities, (2)

handling both data variation and unconstrained properties in one mechanism (e.g. knowing that the object lies to the $-X$ direction of X_0 entails a partial X position constraint of $[-\inf, X_0]$), (3) incremental constraint addition as scene understanding proceeds, (4) a broad range of constraints expressible through algebraic inequalities and (5) the algebraic inequalities maintain the explicit coupling of the relationship between the variables. The main disadvantages seem to be: (1) no representation of the best or most-likely value, (2) difficulty solving the constraints (except when all constraints are linear whereupon the Simplex method is usable), (3) the coupling of uncertainities between variables is not easily exploited (i.e. it often needs more than just intersection of intervals to obtain tight bounds) and (4) uncertainity growth through the composition of uncertainities. Statistical techniques (e.g. [7]) are an attractive alternative representation to intervals (though also having some of these problems), but are not considered further here.

Each type of model-to-data surface feature pairing is given in the subsections below. There are constraints given for each type of pairing, because, for example, the model-based scene analysis program might decide that the curvature of a particular data patch is small, and hence the data patch might be mis-classified as planar. Alternatively, a planar patch might have a small curvature estimated because of noise. The decision about the pairings is the responsibility of the scene analysis program, which then determines which constraints are generated. Hence, constraints for all possible patch matchings are elaborated below.

In each of the subsections below we examine several patch relationships. The main consideration is the extent to which we can constrain their relative position based on *a priori* knowledge. For example, if we are observing rigid man-made identical objects then we can assume that the observed data patch will be a subset of the corresponding model patch. This is our assumption A3, and there are both stronger and weaker assumptions considered. The assumptions (from weak-to-strong) are:

A1 Both model and data patches are part of a larger surface of bounded size R (e.g. the whole surface fits inside a sphere of radius $R/2$), but the patches need not overlap.

A2 The model and data patches must touch or overlap.

A3 The data patch is a subset of the model patch.

A4 The centroids of both the model and data patches correspond to the same point.

Note that each assumption is stricter than its predecessor and so any constraint that applies when A1 holds also applies when, for example, A3 holds. Other assumptions can be developed, such as "The data patch would have exactly the same shape as the model patch if it were completely unobscured and no noise were present.". These seem to generate more complicated, somewhat heuristic constraints, or require substantial image processing to extract the information needed to set up a simpler constraint.

The choice of which assumption to apply is currently up to the model matching software. In our matching investigations, we typically use either assumption A2 or A3, depending on whether rigid man-made or more free-form objects are used.

The position constraints are expressed algebraically, using the prefix transformation T applied to some model feature (e.g. surface normal), and showing the relation the transformed model feature must have with observed data features. Where possible, both rotation and translation constraints are specified. Also some constraints are not as tight as they could be to simplify their formulation.

In general, the constraints are of two forms, with rotation constraints of the form:

$$T\vec{v}_m \cdot \vec{v}_d \geq \tau$$

meaning the transformed model vector $\vec{v}_m$ lies within an angular range of a data vector $\vec{v}_d$. Translation constraints are usually of the form:

$$(T\vec{p}_m - \vec{p}_d) \in I(\vec{r})$$

meaning the vector distance between the transformed model point and a given data point lies within the interval vector [1]. Both τ and $I(\vec{r})$ are constants. The vectors $\vec{v}_m$ and $\vec{p}_m$ are usually constants given in the model, and $\vec{v}_d$ and $\vec{p}_d$ are usually observables. The transform T is usually the desired unknown (i.e. the unknown position).

$I(\vec{r})$ is defined:

If:

$$\vec{r} = (r_x, r_y, r_z)$$

Then:

$$I(\vec{r}) = \{(x,y,z) :\mid x \mid \leq \mid r_x \mid, \mid y \mid \leq \mid r_y \mid, \mid z \mid \leq \mid r_z \mid\}$$

In other words, $I(\vec{r})$ is the rectangular solid of R^3 with principal diagonal corners $\vec{r}$ and $-\vec{r}$. The interval vector expression $I(\vec{a}) + I(\vec{b})$ is defined to have the value:

$$\{\vec{c} + \vec{d} : \vec{c} \in I(\vec{a}), \vec{d} \in I(\vec{b})\}$$

Note that this means that the uncertainity intervals are aligned with the coordinate axes, which may introduce unnecessary uncertainity.

The translation constraints depend on the overlap assumptions and are all based on displacements in three orthogonal directions determined by the surface normal ($\vec{n}_d$), the elongation axis ($\vec{a}_d$) and the cross product ($\vec{c}_d$) of the two:

$$\vec{c}_d = \frac{\vec{n}_d \times \vec{a}_d}{\|\vec{n}_d \times \vec{a}_d\|}$$

If assumption A4 holds, then all translation constraints are of the form:

$$(T\vec{p}_m - \vec{p}_d) \in I(\vec{\gamma})$$

and are not listed further below. ($\vec{\gamma}$ is the isotropic error on the observed point $\vec{p}_d$).

4.1 Planar Data Patch

We assume that the observed data patch curvatures are low enough that it is classified as being planar. It might actually be part of, for example, a cylindrical model patch if the width across the axis were small enough, or if the data segmentation process produced tessellated surfaces.

We propose below some constraints that can be inferred assuming the planar data patch is paired with a planar model patch. The rotation constraints are based on aligning normals and elongation axes. The translation constraints are based on proximity and distance between the designated points on the surface.

The first rotation constraint relates the surface normals, within the observed angular error of ϵ, assuming A1:

$$T\vec{n}_m \cdot \vec{n}_d \geq cos(\epsilon)$$

If A3 holds, we can relate the surface orientations, because the bounding widths may limit the possible rotations of the planar model patch about the data patch. This constraint limits the rotation of the model surface in the plane of the data surface, such that it still fits around the observed data patch:

Let:

$\omega_m(\phi)$ = bounding width (see Figure 2) of model surface at orientation ϕ

$\omega_d(\phi)$ = bounding width of the data surface at orientation ϕ

$\Omega = \{\theta : \forall\phi(\omega_m(\phi - \theta) \geq \omega_d(\phi) - \delta_\omega\}$ is the set of possible rotations of the model that can fit around the data

$\vec{a}_m$ = the nominal unit vector in the plane of the model surface (at orientation ϕ_m)

$\vec{a}_d$ = average orientation of $\vec{a}_m$ over Ω (at orientation ϕ_d)

If $size(\Omega)/2 + \beta < \pi$ then:

$$T\vec{a}_m \cdot \vec{a}_d \geq cos(size(\Omega)/2 + \beta)$$

If more than one interval of angles exists in Ω, then each interval creates an independent alternative constraint. (Note that a more complex constraint could be developed that limits the range of $\vec{a}_m$ to lying in the plane, rather than as formulated here.)

All translation constraints have the form:

$$(T\vec{p}_m - \vec{p}_d) \in I(A\vec{n}_d) + I(B\vec{a}_d) + I(C\vec{c}_d) + I(\vec{\gamma})$$

where the constants A, B and C depend on the specific assumptions as given below. If assumption A1 or A2 holds, then the orientation of $\vec{a}_d$ is arbitrary, and is chosen to correspond with angle $\phi_d = 0$. Let:

$$M = min_\phi(\omega_m(\phi))$$

$$N = max_\phi(\omega_m(\phi))$$

Then, the maximum width of the model in orientations compatible with $\vec{a}_d$ is:

$$U = max_\phi(\omega_m(\phi)) \text{ over } \phi_d - \phi \in \Omega$$

and the maximum width at right angles is:

$$V = max_\phi(\omega_m(\phi + \pi/2)) \text{ over } \phi_d - \phi \in \Omega$$

Asmp.	A	B	C
A1	$\epsilon\sqrt{B^2 + C^2}$	$R - \frac{1}{2}(M + \omega_d(\phi_d)) - \delta_\omega)$	$R - \frac{1}{2}(M + \omega_d(\phi_d + \pi/2) - \delta_\omega)$
A2	$\epsilon\sqrt{B^2 + C^2}$	$\frac{1}{2}(N + \omega_d(\phi_d) + \delta_\omega)$	$\frac{1}{2}(N + \omega_d(\phi_d + \pi/2) + \delta_\omega)$
A3	$\epsilon\sqrt{B^2 + C^2}$	$\frac{1}{2}(U - \omega_d(\phi_d) + \delta_\omega)$	$\frac{1}{2}(V - \omega_d(\phi_d + \pi/2) + \delta_\omega)$

We illustrate this rotation constraint with an example. Assume that we have the model and data patches given in Figure 7. For the sake of simplicity, assume that the figures both lie in the plane of the paper, so that their surface normals point facing the viewer. The nominal model direction vector ($\vec{a}_m$ is shown along with the patch width ($\omega_m(0)$) assuming that the vector corresponds with orientation $\phi_m = 0$. The data patch width ($\omega_d(\phi_d)$) is shown for another orientation $\phi_d = \pi/4$.

A sketch of the two functions $\omega_m(\phi)$ and $\omega_d(\phi)$ is shown next. Suppose that $\omega_m(0) = 100$, $\omega_m(\pi/4) = 50$ and $\omega_m(\pi/2) = 30$. Let $\omega_d(\phi_d) = 50$ and $\delta_\omega = 0$. From a casual inspection of the two functions, Ω is approximately $[0, \frac{\pi}{2}]$. This implies that the second rotation constraint is approximately:

$$T\vec{a}_m \cdot \vec{a}_d \geq cos(\frac{\pi}{4} + \beta)$$

If assumption A4 holds, then a similar constraint can be developed, except where $\omega(\phi)$ refers to the cross-section width about the patch centers. Then, these values imply that the example above produces the following constraint parameters: $M = 100$, $N = 30$, $U = 100$, $V = 50$ and hence $B = \frac{1}{2}(100 - 50 + 0) = 25$, $C = \frac{1}{2}(50 - 15 + 0) = 17.5$ and $A = 30.5\epsilon$.

We record the results for matching with non-planar model patches in Appendix A.

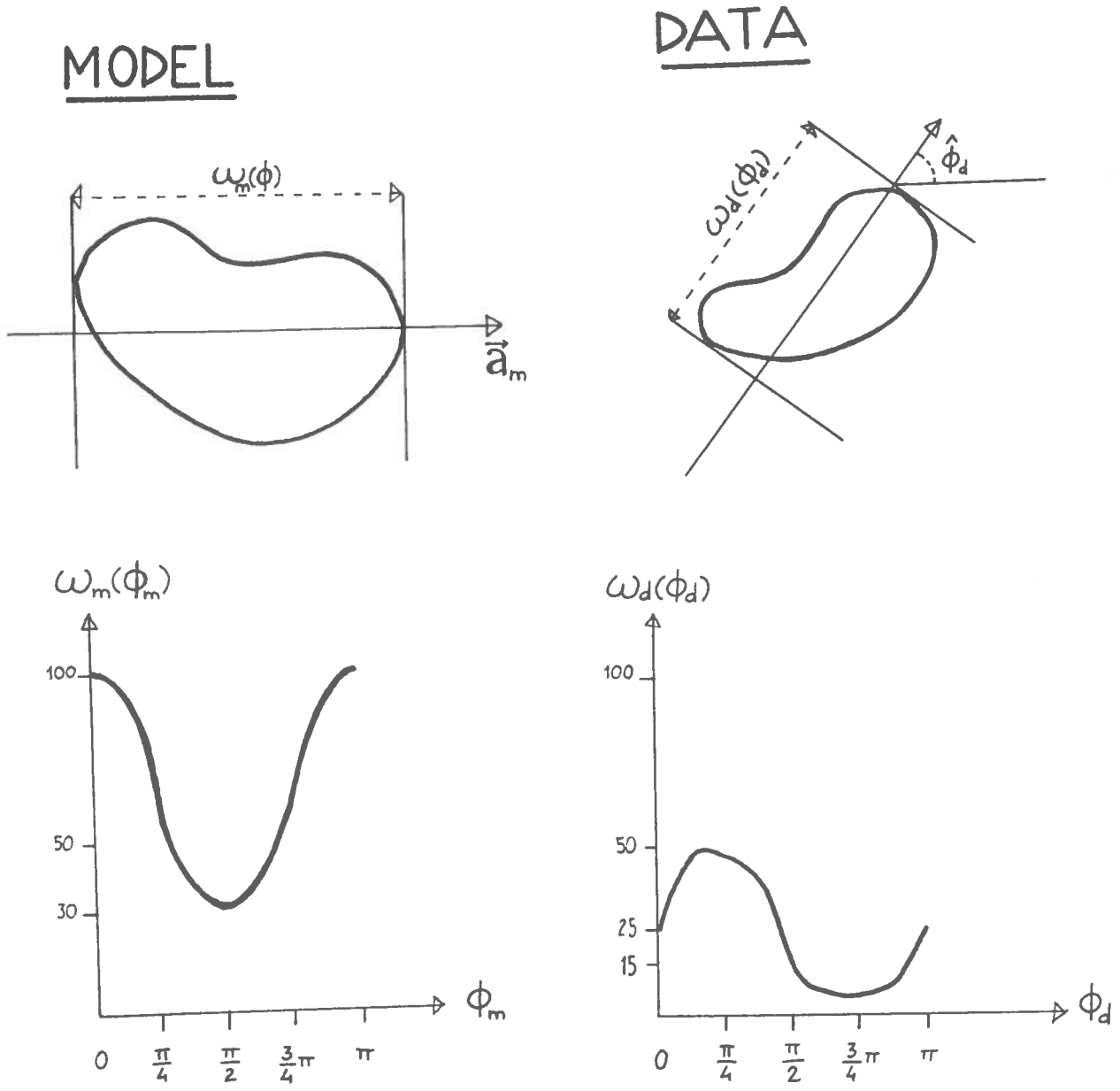


Figure 7: Plane-Plane Constraint Example

4.2 Cylindrical Data Patch

We assume that the observed data patch curvatures are such that the patch is classified as cylindrical. It might be part of, for example, a generic model patch if the portion across the second axis were small enough, or if the data segmentation process produced unexpected results.

We propose below some constraints that can be inferred assuming the cylindrical data patch is paired with either a partial or complete cylindrical model patch. The rotation constraints are based on aligning normals and curvature axes. The translation constraints are based on proximity and distance between designated points on the surface.

4.2.1 Cylindrical Data Patch - Cylindrical Model Patch

We first consider the constraints derivable assuming that the model patch is cylindrical, but does not extend fully around the cylinder. The first rotation constraint relates the curvature axes, within the observed angular error of β:

$$T\vec{a}_m \cdot \vec{a}_d \geq cos(\beta)$$

Two independent constraints are generated for the two possible orientations of the elongation axis $\vec{a}_d$.

A second constraint relates the surface normal vectors, depending on the surface relationship assumptions (i.e. how much the patches overlap affects how closely the normal's can be related). This constraint limits the orientation of the model surface relative to the cylindrical data surface. If $\tau \leq \pi$, the following constraint can be added:

$$T\vec{n}_m \cdot \vec{n}_d \geq cos(\tau)$$

where:

Assumption	τ
A1	$R\kappa_m - \frac{1}{2}\theta_m - \frac{1}{2}\theta_d + \epsilon$
A2	$\frac{1}{2}(\theta_m + \theta_d) + \sigma + \epsilon$
A3	$\frac{1}{2}(\theta_m - \theta_d) + \sigma + \epsilon$
A4	ϵ

The translation constraints depend on the overlap assumptions, given that the data match must lie somewhere along the cylinder of which the model patch is part. All are based on displacements in three orthogonal directions determined by the surface normal ($\vec{n}_d$), the cylinder axis ($\vec{a}_d$) and the cross product ($\vec{c}_d$) of the two. Then, the position constraints have the form:

$$(T\vec{p}_m - \vec{p}_d + E\vec{n}_d) \in I(A\vec{n}_d) + I(B\vec{a}_d) + I(C\vec{c}_d) + I(\vec{\gamma})$$

$$A = E + \epsilon\sqrt{B^2 + C^2}$$

where the constants B, C and E depend on the specific assumptions as given below.

Asmp.	B	C	E
A1	$R - \frac{1}{2}(l_m + l_d + \delta_l)$	$\frac{1}{\kappa_m}$	$\frac{1}{\kappa_m}$
A2	$\frac{1}{2}(l_m + l_d + \delta_l)$	$\frac{1}{\kappa_m} sin(\frac{1}{2}(\theta_m + \theta_d) + \sigma))$	$\frac{1}{2\kappa_m}(1 - cos(\frac{1}{2}(\theta_m + \theta_d) + \sigma))$
A3	$\frac{1}{2}(l_m - l_d + \delta_l)$	$\frac{1}{\kappa_m} sin(\frac{1}{2}(\theta_m - \theta_d) + \sigma))$	$\frac{1}{2\kappa_m}(1 - cos(\frac{1}{2}(\theta_m - \theta_d) + \sigma))$

4.2.2 Cylindrical Data Patch - Model Cylinder

Here we assume that the model patch is a complete cylinder. The rotation constraint relates the curvature axes within the observed angular error of β:

$$T\vec{a}_m \cdot \vec{a}_d \geq cos(\beta)$$

Two independent alternative constraints are generated for the two possible orientations of the elongation axis $\vec{a}_d$.

The translation constraints depend on the overlap assumptions, given that the data patch must lie somewhere along the model cylinder. All are based on displacements in

three orthogonal directions determined by the surface normal ($\vec{n}_d$), the cylinder axis ($\vec{a}_d$) and the cross product ($\vec{c}_d$) of the two. Then, the position constraints have the form:

$$(T\vec{p}_m - \vec{p}_d + \frac{1}{\kappa_m}\vec{n}_d) \in I(A\vec{n}_d) + I(B\vec{a}_d) + I(C\vec{c}_d) + I(\vec{\gamma})$$

where the constants A, B and C depend on the specific assumptions as given below.

Asmp.	A	B	C
A1	$\epsilon\sqrt{B^2+C^2}$	$R - \frac{1}{2}(l_m + l_d + \delta_l)$	$\frac{1}{\kappa_m}sin(\epsilon) + Rsin(\beta)$
A2	$\epsilon\sqrt{B^2+C^2}$	$\frac{1}{2}(l_m + l_d + \delta_l)$	$\frac{1}{\kappa_m}sin(\epsilon) + \frac{1}{2}(l_m + l_d)sin(\beta)$
A3	$\epsilon\sqrt{B^2+C^2}$	$\frac{1}{2}(l_m - l_d + \delta_l)$	$\frac{1}{\kappa_m}sin(\epsilon) - \frac{1}{2}(l_m + l_d)sin(\beta)$

We record the results for matching with non-cylindrical model patches in Appendix B.

4.3 Conical Data Patch

We now assume that the observed data patch curvatures are such that it is classified as being conical.

This section gives the constraints derivable assuming the conical data patch is paired with a conical model patch. The rotation constraints are based on aligning normals and curvature axes. The translation constraints are based on proximity and distance between the designated points on the surface.

The first rotation constraint relates the cone axes, within the observed angular error of β:

$$T\vec{a}_m \cdot \vec{a}_d \geq cos(\beta)$$

Here, only one constraint is generated, because the orientation of the cone axis can be deduced from the data.

A second constraint relates the surface normal vectors, depending on the surface relationship assumptions. This constraint limits the rotation of the model surface around the conical data surface, and is similar to the cylinder-cylinder constraint, except that the normals are not diametrically opposed when on opposite sides of the cone. The constraints

are of the form:

$$T\vec{n}_m \cdot \vec{n}_d \geq \tau$$

where:

Assumption	τ
A1	$\frac{\rho_m^2-1}{\rho_m^2+1}$
A2	$\frac{1}{\rho_m^2+1}(cos(\frac{1}{2}(\theta_m+\theta_d)+\sigma+\epsilon)+\rho_m^2)$
A3	$\frac{1}{\rho_m^2+1}(cos(\frac{1}{2}(\theta_m-\theta_d)+\sigma+\epsilon)+\rho_m^2)$
A4	$\frac{1}{\rho_m^2+1}(cos(\epsilon)+\rho_m^2)$

The translation constraint is simpler, in that it does not depend on the overlap assumptions, because the tip of the cone is a fixed point deducible from both the model and data. The constraint is based on displacements in three orthogonal directions determined by the surface normal ($\vec{n}_d$), the cone axis ($\vec{a}_d$) and the cross product ($\vec{c}_d$) of the two. Then, the position constraint has the form:

$$(T\vec{t}_m - \vec{p}_d + \frac{\sqrt{E}}{\kappa_d}\vec{n}_d + \frac{E}{\rho_m\kappa_d}\vec{a}_d) \in I(A\vec{n}_d) + I(B\vec{a}_d) + I(C\vec{c}_d) + I(\vec{\gamma})$$

$$E = \rho_m^2 + 1$$

where the constants A, B and C are:

A	B	C
$\alpha\sqrt{E} + \frac{\beta E}{\rho_m\kappa_d}$	$\frac{\epsilon\sqrt{E}}{\kappa_d} + \frac{\alpha E}{\rho_m}$	$\frac{\epsilon\sqrt{E}}{\kappa_d} + \frac{\beta E}{\rho_m\kappa_d}$

We record the results for matching with non-conical model patches in Appendix C.

4.4 Generic Data Patch

We assume that the observed data patch curvatures are such that it is classified as being generic; that is, it has two non-zero curvatures. The curvatures may be either positive or

negative. The data patch might be part of, for example, a cylindrical model patch if the portion along the axis were small enough that the noise gave the impression of curvature.

Here we give the constraints that can be generated assuming the generic data patch is paired with a generic model patch. The rotation constraints are based on aligning normals and curvature axes. The translation constraints are based on proximity and distance between the designated points on the surface.

There are three rotation constraints that relate the curvature axes and the observed normal. Because there are two directions of curvature, it is possible for there to be four distinct orientations for the surface (2 pairings and 2 orientations of the curvature axes) at approximately $\pi/2$ rotations about the normal. However, it may be possible to eliminate two of the pairings by showing that the signs or magnitudes of the corresponding curvatures are incompatible. In the constraints given below, it is assumed that the pairings have already been made.

The first constraint relates the surface normal vectors, depending on the surface relationship assumptions. If $\tau_1 \leq \pi$, the following constraint can be added:

$$T\vec{n}_m \cdot \vec{n}_d \geq cos(\tau_1)$$

where:

Assumption	τ_1
A1	$R\kappa_{1m} + R\kappa_{2m} - \frac{1}{2}(\theta_{1m} + \theta_{1d} + \theta_{2m} + \theta_{2d}) + \epsilon$
A2	$\frac{1}{2}(\theta_{1m} + \theta_{1d}) + \frac{1}{2}(\theta_{2m} + \theta_{2d}) + \sigma_1 + \sigma_2 + \epsilon$
A3	$\frac{1}{2}(\theta_{1m} - \theta_{1d}) + \frac{1}{2}(\theta_{2m} - \theta_{2d}) + \sigma_1 + \sigma_2 + \epsilon$
A4	ϵ

The second constraint relates one curvature axis, depending on the surface relationship assumptions. If $\tau_2 \leq \pi$, the following constraint can be added:

$$T\vec{a}_{1m} \cdot \vec{a}_{1d} \geq cos(\tau_2)$$

where:

Assumption	τ_2
A1	$R\kappa_{2m} - \frac{1}{2}(\theta_{2m} + \theta_{2d}) + \beta_1$
A2	$\frac{1}{2}(\theta_{2m} + \theta_{2d}) + \sigma_2 + \beta_1$
A3	$\frac{1}{2}(\theta_{2m} - \theta_{2d}) + \sigma_2 + \beta_1$
A4	β_1

The third constraint relates the other curvature axis, depending on the surface relationship assumptions. If $\tau_3 \leq \pi$, the following constraint can be added:

$$T\vec{a}_{2m} \cdot \vec{a}_{2d} \geq cos(\tau_3)$$

where:

Assumption	τ_3
A1	$R\kappa_{1m} - \frac{1}{2}(\theta_{1m} + \theta_{1d}) + \beta_2$
A2	$\frac{1}{2}(\theta_{1m} + \theta_{1d}) + \sigma_1 + \beta_2$
A3	$\frac{1}{2}(\theta_{1m} - \theta_{1d}) + \sigma_1 + \beta_2$
A4	β_2

The translation constraints depend on the overlap assumptions, given that the data match must lie somewhere along the generic surface of which the model patch is part. All are based on displacements in three orthogonal directions determined by the surface normal ($\vec{n}_d$) and the two curvature axes ($\vec{a}_{1d}$, $\vec{a}_{2d}$). Then, the position constraints have the form:

$$(T\vec{p}_m - \vec{p}_d + D\vec{n}_d) \in I(A\vec{n}_d) + I(B\vec{a}_{1d}) + I(C\vec{a}_{2d}) + I(\vec{\gamma})$$

where the constants A, B, C and D depend on the specific assumptions as given below.

Asmp.	A	B
A1	$D + \epsilon\sqrt{B^2 + C^2}$	$\frac{1}{\kappa_{2m}}$
A2	$D + \epsilon\sqrt{B^2 + C^2}$	$\frac{1}{\kappa_{2m}} sin(\frac{1}{2}(\theta_{2m} + \theta_{2d}) + \sigma_2)$
A3	$D + \epsilon\sqrt{B^2 + C^2}$	$\frac{1}{\kappa_{2m}} sin(\frac{1}{2}(\theta_{2m} - \theta_{2d}) + \sigma_2)$

Asmp.	C
A1	$\frac{1}{\kappa_{1m}}$
A2	$\frac{1}{\kappa_{1m}} sin(\frac{1}{2}(\theta_{1m} + \theta_{1d}) + \sigma_1)$
A3	$\frac{1}{\kappa_{1m}} sin(\frac{1}{2}(\theta_{1m} - \theta_{1d}) + \sigma_1)$

Asmp.	D
A1	$\frac{1}{2}(\frac{1}{\kappa_{1m}} + \frac{1}{\kappa_{1m}})$
A2	$\frac{1}{2\kappa_{1m}}(1 - cos(\frac{1}{2}(\theta_{1m} + \theta_{1d}) + \sigma_1)) + \frac{1}{2\kappa_{2m}}(1 - cos(\frac{1}{2}(\theta_{2m} + \theta_{2d}) + \sigma_2))$
A3	$\frac{1}{2\kappa_{1m}}(1 - cos(\frac{1}{2}(\theta_{1m} - \theta_{1d}) + \sigma_1)) + \frac{1}{2\kappa_{2m}}(1 - cos(\frac{1}{2}(\theta_{2m} - \theta_{2d}) + \sigma_2))$

We record the results for matching with non-generic model patches in Appendix D.

5 Evaluating The Position Constraints

The constraints presented above hold irrespective of the method used to evaluate them. However, it is important to show that the constraints can be solved to provide position estimates for the vision system. This section shows one method for evaluating the constraints, but before we do this, we shall discuss some other approaches used for position estimation.

There are many (mainly statistical) approaches to solving for 3D positions from 2D and 3D data, based on minimizing an error function of the observed data and transformed model features. A few examples from the Computer Vision literature follow, although there are also considerable results from photogrammetry research. From two dimensional data, Roberts [22] used a heuristic technique based on deriving values of the homogeneous coordinate matrix, for perspective projection and assuming point correspondences. ACRONYM [4] used symbolic algebra to constrain the range of values that the position

parameters could take, where the constraints were derived from algebraic relations between 2D image and 3D scene sizes and positions. Lowe [20] used both point and line feature correspondences, but solved for the positions using an iterative search technique. Haralick *et al* [16] described and demonstrated the performance of several robust statistical methods for both estimating positions from both two and three dimensional feature point correspondences.

From three dimensional data, the work of Faugeras and Hebert [8] is highly regarded. Assuming three dimensional vector and point correspondences, they used an eigenvalue technique to solve for the quaternion rotation and vector translation, with the desired transformation minimizing the distance between the model and observed features (e.g. vectors). Grimson and Lozano-Perez [15] used pairs of model-to-data vector correspondences to directly solve for the rotations (and then clustered the results to reduce error). The essence of their method is very similar to the approach used in the TV2 module described below. Ballard and Tanaka [2] described a network-based geometric reasoning engine. Their implementation used nodes representing values of transformation parameters linking image and model features. In a sense, this is a discrete implementation of the geometric reasoning embedded in the network processes described below.

The above methods were largely based on independent measurements of point and vector quantities. Exploiting local geometry can sometimes allow easy determination of position, such as when using 2D projections of trihedral vertices [18] or coplanar conics [24]. However, we are interested in the more abstract formulation, represented only as point and vector correspondences.

Moreover, the constraints presented in Section 4 were based on information extracted from a single patch correspondence, whereas the other methods require more correspondences. We do not claim that this is better, rather we claim that our method defines additional constraints that can also be exploited, particularly when reduced information is available.

There is then the problem of solving the constraints over the intervals to estimate positions. The most famous method is that used in ACRONYM [4], based on symbolic algebra, but other methods can also be used (e.g. [23]). We have been using a value

passing network to solve the constraints [12],[13]. The networks produce tighter bounds than simply using symbolic algebra and have greatly improved efficiency.

The networks are created from an underlying algebraic description of three-dimensional geometric relations and transformations. These networks implement the five key visual geometric reasoning functions: locate (calculate position from model-to-data pairings), predict (predict data position from model and estimated position), transform (composition of reference frame transformations), inverse (invert a transformation) and merge (reconcile two or more transformations) [21]. The construction and evaluation of these networks was described in [12], but a quick overview is given now.

The basis of the network is the propagation of updated bounds, through functional units linked according to the algebraic problem specification. A simple example is based on the inequality:

$$A \leq B - C$$

By the SUP/INF calculus (e.g. [4]), the upper bound of A is constrained by:

$$SUP(A) \leq SUP(B) - INF(C)$$

as are the lower bound of B and upper bound of C:

$$INF(B) \geq INF(A) + INF(C) \quad SUP(C) \leq SUP(B) - INF(A)$$

Thus, one can use the value of $SUP(B) - INF(C)$ as an estimated upper bound for A, etc. These relationships are used to create the network for this single inequality, which is shown in Figure 8. As new bounds on B are computed, perhaps from other relationships, they propagate through the network to help compute new bounds on A and C.

There are two advantages to the network structure. First, because values propagate, local improvements in estimates propagate to help constrain other values elsewhere. Second, these networks have a natural wide-scale parallel structure (e.g. 1000+) that might eventually lead to extremely fast evaluation.

For a given problem, the networks can be complicated, particularly since there may be both exact and heuristic bounding relationships. For example, the network expressing the composition of two 3D reference frame transformations to give a third contains about 2000

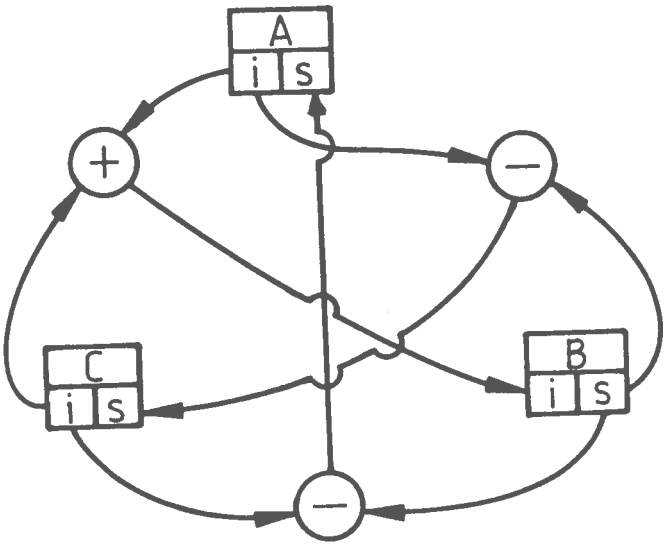


Figure 8: Small Constraint Network Example

function nodes (of types '+', '-', '*', '/', 'sqrt', 'max', '≥', 'if', etc.). Though the network is formulated for parallel evaluation, simulated serial evaluation is also fast, because small changes are truncated to prevent trivial propagations, unlike other constraint propagation approaches (e.g. [6]). Convergence is guaranteed (or inconsistency detected) because bounds can only tighten (or cross) and there must be a minimum change for propagation to occur.

The creation of the networks is time-consuming, requiring a symbolic analysis of the algebraic inequalities. Fortunately, there is a natural modular structure arising from the types of problems encountered during scene analysis, where most geometric constraints are of the type described above. Hence, it is possible to pre-compile network modules for each relationship, and merely connect a new instance of the module into the network as scene analysis proceeds. To date, we have identified and implemented network modules for:

SS - two scalars are close in value

PP - two points are close in location

VV - two vectors point in nearly the same direction

DOT3 - the dot product of two 3-vectors exceeds a scalar

DOT3N - the absolute value of the dot product of two 3-vectors is less than a scalar

UNIT3 - enforcing a unit 3-vector constraint

UNIT4 - enforcing a unit 4-vector constraint

TP - a transformation links a pair of points

TV - a transformation links a pair of vectors

TV2 - a transformation links two pairs of vectors

TT - a transformation maps from one position to a second position
by a third relative position

P2V - a vector can be defined by two points

QWθ - a quaternion is equivalent to an axis and an angle

For example, the **UNIT3** module is defined by the following constraints between the components of a vector (p_x, p_y, p_z):

$$p_i = (1 - p_j^2 - p_k^2)/p_i$$
$$| p_i | \leq \sqrt{1 - p_j^2 - p_k^2}$$
$$| p_i | \leq 1.7321 - | p_j | - | p_k |$$
$$| p_i | \leq 1.4143 - | p_j |$$
$$| p_i | \leq 1$$

The first two relationships are exact, and the last three give correct, but weaker bounds on the given quantities. The weaker bounds are useful because (1) it is not always possible for the network to use the exact bounds (e.g. when dividing by an interval containing zero) and (2) the initial bounds are sometimes so unconstrained that the weaker relationships provide better bounding. Other weaker bounds and formulations of the relationships are possible, and could be included in a network module.

We now include a simple example of network use. Suppose subcomponents B and C are rigidly connected to form object A. Given the estimated positions of the subcomponents in

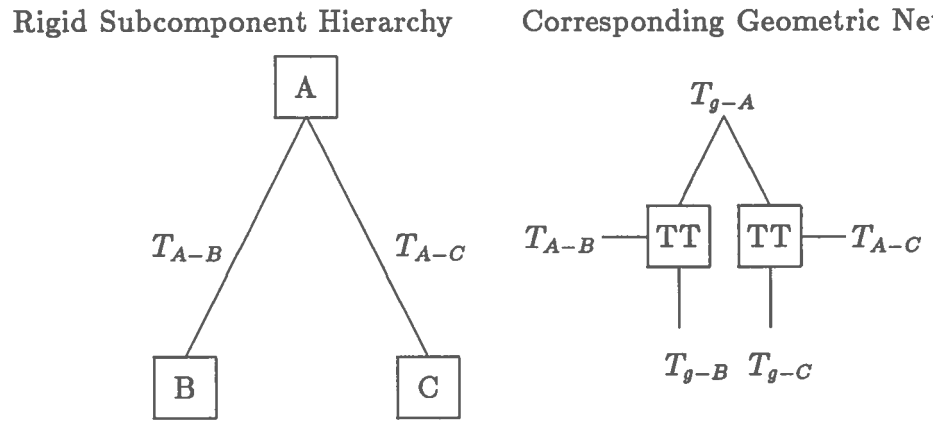


Figure 9: A Simple Geometric Reasoning Network

the global coordinate system, T_{g-B} and T_{g-C}, and the transformations between the object and subcomponent coordinate systems, T_{A-B} and T_{A-C}, then we can estimate the global object position, T_{g-A}, using two instances of the "TT" module listed above. Figure 9 shows this network. Notice that each subcomponent gives an independent estimate of T_{g-A}, so that the network keeps the tightest bounds on each component of the position. Any tighter resulting bounds then propagate back through the modules to refine the subcomponent position estimates.

6 Performance Analysis Of The Constraints

A great many experiments can be set up to use the constraints presented above, but it is not useful to demonstrate all of them. This section will present results from the four main patch pairing relationships:

1. planar model patch to planar data patches,
2. cylindrical model patch to cylindrical data patches,
3. conical model patch to conical data patches and
4. generic model patch to generic data patches

for both Assumptions A2 (overlap) and A3 (data subset of model). These are justified because the segmentation and model-matching should correctly extract and pair most data patches, otherwise it is unlikely that recognition can proceed. It is also useful to know the "best case" performance. Assumption A2 often occurs with man-made parts, where all features are regular and hence data features are often a subset of the data present. Assumption A3 often occurs with more natural objects, where the data patch need not be contained in the model patch, but will usually have some overlap.

The experiments will involve the estimation of the reference frame of a single data patch, given its observation, and applying the constraints given in Section 4.

The experiments start with a known model surface normal ($\vec{n}_m = (0,0,-1)$), axis ($\vec{a}_m$ or $\vec{a}_{1m} = (1,0,0)$ for the generic patch) and central point ($\vec{p}_m = (0,0,0)$).

We then randomly pick a rotation quaternion (distributed uniformly over the 4-dimensional unit sphere) and a translation (each component distributed uniformly over the range ± 100). Using this position (P_{true}), we transform the input features to derive some scene features: $\vec{n}_s$, $\vec{a}_s$ and $\vec{p}_s$.

The scene features are then corrupted with randomly generated noise to generate the observed features: $\vec{n}_d$, $\vec{a}_d$ and $\vec{p}_d$. The corruptions were uniformly distributed subject to:

$$\vec{n}_s \cdot \vec{n}_d \geq cos(\epsilon)$$

$$\vec{a}_s \cdot \vec{a}_d \geq cos(\beta)$$

$$(\vec{p}_s - \vec{p}_d) \in I(\vec{\gamma})$$

The observed features were then paired with the modeled features, to derive the geometric position constraints, based on the results of Sections 4.1, 4.2.1, 4.3 and 4.4. The constraints depended on the assumptions made and the extent of the patches. Here, we used assumptions A2 and A3, as discussed in the experiment details below.

The constraints were input into a geometric reasoning network [12] set up for evaluating these constraints. Figure 10 shows the network for the plane-plane, cylinder-cylinder and cone-cone experiments described below. (The generic-generic network is similar, but slightly more complicated.) This network implements a SUP/INF interval bounding of all

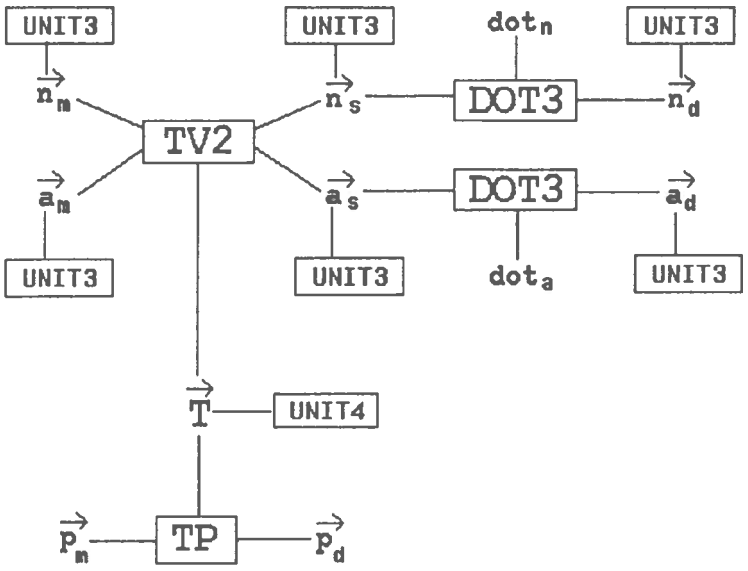


Figure 10: Geometric Reasoning Network For Estimating Position

components of the position, as a function of the intervals bounding the inputs. The inputs for this network are:

- each component of the model vectors $\vec{n}_m$, $\vec{a}_m$ and $\vec{p}_m$ extended to form intervals with width 0.0001 (a small value, as each quantity needs to be an interval)
- the data vectors $\vec{n}_d$ and $\vec{a}_d$ extended to form intervals with width 0.0001,
- the estimated position $\vec{p}_d$ extended to form a larger interval according to the translation constraints given in Section 4 and
- the rotation dot product constraints dot_n and dot_a given by the rotation constraints

$$T\vec{n}_m \cdot \vec{n}_d \geq dot_n$$

$$T\vec{a}_m \cdot \vec{a}_d \geq dot_a$$

The main component of the network is the "TV2" module near the center. This module relates the input and output vectors according to the position $\vec{P}$ (near the bottom left) and is defined and created from the algebraic relationships between the input and output

vectors. Similarly, the translation component of $\vec{P}$ is estimated by the "TP" module that relates the model and observed feature points. There are also two "DOT3" modules that implement the rotation dot product constraints. The vectors $\vec{n}_i$ and $\vec{a}_i$ are intermediate results. Finally, there are several "UNIT3" and "UNIT4" modules that enforce unit 3-vector and 4-vector constraints on the various vectors.

This sets up the network for estimating the position $\vec{P}$. We then evaluate the network and extract the estimate position.

The relationship between the true position and the estimated position provides the statistical output described below. All the reported results below are based on 100 randomly generated trials of the particular experiment. The data from the same 100 trials was used for each value of the percentage overlap parameter, to compare the results.

The first experiment is for the model plane - data plane case. The following parameters were used (where p is a percentage parameter described below):

Model Value		Data Value		Data Uncertainity	
$\omega_m(\phi)$	*	$\omega_d(\phi)$	$p\omega_m(\phi)$	δ_ω	1.0
$\vec{n}_m$	(0,0,-1)	$\vec{n}_d$	transformed	ϵ	0.1
$\vec{a}_m$	(1,0,0)	$\vec{a}_d$	transformed	β	0.1
$\vec{p}_m$	(0,0,0)	$\vec{p}_d$	transformed	γ	1.0

* - the test figure was a rectangle of length 100 and width 30

The experiment was run using assumptions A2 and A3 and the parameter p was allowed to vary from 1.0 to 0.1 in steps of 0.1. This allowed the exploration of the constraint obtainable because the patch was limited in its ability to rotate relative to the model patch. In general, we can see below that as the size of the data patch tends to zero, the two assumptions give nearly the same results, as is expected (because the effective translation constraints become the same).

In all experiments, the true object position was contained within the estimated interval position, so the constraints appear to be correct. There is then the question of how tight the constraints are, and this is the main point of the experiments.

Five statistics were extracted from the experiments:

1. the number of times that the rotation had at least two independent constraints and hence was fully constrained (though subject to error).

2. the mean angle between the input and output quaternion vectors (a measure of how well the rotation was estimated, and ideally should be zero).

3. the mean width of the interval of possible values for a component of the rotation quaternion (a measure of how tight the estimates are). The geometric mean is used because it is a measure of the volume of the uncertainity.

4. the mean distance between the input and estimated translation (a measure of how well the translation was estimated, and ideally should be zero).

5. the average width of the interval of possible values for a component of the translation (a measure of how tight the estimates are). The geometric mean is used again because it is a measure of the volume of the uncertainity.

More precisely, the error measures are defined by:

Let:

$$Angle = \frac{1}{N}\sum_{i=1}^{N} cos^{-1}(\hat{d}_{0i}q_{0i} + \hat{d}_{1i}q_{1i} + \hat{d}_{2i}q_{2i} + \hat{d}_{3i}q_{3i})$$

$$Quat\_Width = (\frac{1}{N}\sum_{i=1}^{N} width(D_{0i}) * width(D_{1i}) * width(D_{2i}) * width(D_{3i}))^{\frac{1}{4}}$$

$$Trans = \frac{1}{N}\sum_{i=1}^{N} \sqrt{(t_{xi} - v_{xi})^2 + (t_{yi} - v_{yi})^2 + (t_{zi} - v_{zi})^2}$$

$$Trans\_Width = (\frac{1}{N}\sum_{i=1}^{N} width(T_{xi}) * width(T_{yi}) * width(T_{zi}))^{\frac{1}{3}}$$

where

q_{ji} is the jth component of the ith true scalar rotation quaternion $\vec{q}_i$

d_{ji} is the mean value of the jth component (D_{ji}) of the ith estimated rotation interval quaternion $\vec{D}_i$

$\hat{d}_{ji}$ is the unit-quaternion normalized value of d_{ji}

t_{xi} is the mean value of the x-component (T_{xi}) of the ith estimated translation interval vector $\vec{T}_i$

Table 1: Plane-Plane Parameter Estimation Statistics for Assumption A3

p (%)	Constr.	Angle	Quat. Width	Trans.	Trans. Width
100	100	0.23	0.69	1.01	3.82
90	100	0.27	0.74	1.01	12.74
80	100	0.31	0.77	1.01	22.75
70	100	0.41	0.89	1.01	32.80
60	100	0.46	0.95	1.01	44.52
50	100	0.53	1.04	1.01	56.83
40	100	0.77	1.25	1.01	72.26
30	100	1.07	1.48	1.01	92.86
20	0	1.42	1.52	1.01	100.28
10	0	1.42	1.52	1.01	107.20

Table 2: Plane-Plane Parameter Estimation Statistics for Assumption A2

p (%)	Constr.	Angle	Quat. Width	Trans.	Trans. Width
100	0	1.42	1.52	1.01	182.89
90	0	1.42	1.52	1.01	176.03
80	0	1.42	1.52	1.01	169.17
70	0	1.42	1.52	1.01	162.31
60	0	1.42	1.52	1.01	155.44
50	0	1.42	1.52	1.01	148.57
40	0	1.42	1.52	1.01	141.69
30	0	1.42	1.52	1.01	134.81
20	0	1.42	1.52	1.01	127.92
10	0	1.42	1.52	1.01	121.02

v_{xi} is the x-component of the ith true translation $\vec{v}_i$

$width(X)$ is the width of scalar interval X

The results of the experiment for assumption A3 are shown in Table 1, where the angle is expressed in radians and the translation is in arbitrary units. As the shrinkage percentage was increased, the rotation and translation errors increased, as expected, because the model patch was freer to move about the data patch. At 20% there was a dramatic increase in the errors, because the patch size was small enough that it could rotate freely within the model patch, and hence only one direction vector was effectively constrained (the surface

normal). The mean translation error was constant and mainly resulted from the isotropic error on the observed central point.

When assumption A2 was used, the results in Table 2 were obtained. Here, only one rotation constraint was obtained (that of the aligned normals). This explained the constant large rotation error. As before, the translation error was constant and small. The translation error range started larger, because of the larger area over which the model surface could move and still have overlap with the data surface. However, unlike the previous case (i.e. using assumption A3), as the percentage of the observed surface decreased, the translation error also decreased, because the model surface was required to overlap with the smaller data surface.

When the experiments were run for the cylinder-cylinder case, the experimental setup was similar to the plane-plane case, except here the axis is always well constrained, whereas the normal is not so. A model cylinder patch of length 100, radius 20 and angular width 1.0 was used. The linear length and angular width were reduced by fixed percentages and the position estimated. The error tolerances were:

Model Value		Data Value		Data Uncertainity	
l_m	100	l_d	$100p$	δ_l	1.0
θ_m	1.0	θ_d	$1.0p$	σ	0.1
κ_m	0.05	not used	not used	not used	not used
$\vec{n}_m$	(0,0,-1)	$\vec{n}_d$	transformed	ϵ	0.1
$\vec{a}_m$	(1,0,0)	$\vec{a}_d$	transformed	β	0.1
$\vec{p}_m$	(0,0,0)	$\vec{p}_d$	transformed	γ	1.0

The same statistics were obtained as for the plane-plane case and the experiments were run for both assumptions A2 and A3. The results for the experiments are shown in Tables 3 and 4.

Under assumption A3, the surface normal was progressively less constrained, whereas the axis was always estimated well. Since the normal was always partly constrained, this meant that there was no catastrophic breakdown in rotation estimation as the patch became small, but the mean and interval errors increased smoothly as the data patch size

Table 3: Cylinder-Cylinder Parameter Estimation Statistics for Assumption A3

p (%)	Constr.	Angle	Quat. Width	Trans.	Trans. Width
100	100	0.36	0.79	1.02	5.75
90	100	0.42	0.89	1.03	12.25
80	100	0.46	0.94	1.04	18.64
70	100	0.48	0.97	1.06	25.02
60	100	0.50	1.02	1.10	31.40
50	100	0.56	1.07	1.16	37.79
40	100	0.61	1.12	1.25	44.18
30	100	0.64	1.17	1.37	50.56
20	100	0.69	1.22	1.53	56.95
10	100	0.77	1.26	1.72	63.33

Table 4: Cylinder-Cylinder Parameter Estimation Statistics for Assumption A2

p (%)	Constr.	Angle	Quat. Width	Trans.	Trans. Width
100	100	1.13	1.48	5.53	132.41
90	100	1.12	1.48	5.10	126.27
80	100	1.11	1.47	4.68	120.10
70	100	1.09	1.47	4.27	113.89
60	100	1.07	1.46	3.88	107.64
50	100	1.04	1.44	3.51	101.38
40	100	1.01	1.42	3.16	95.08
30	100	0.99	1.41	2.82	88.77
20	100	0.95	1.38	2.51	82.42
10	100	0.84	1.32	2.22	76.08

decreased.

Under assumption A2, the rotation was still always constrained, though not as well as under assumption A3, because the axis is observable and the normal is partly constrained. Here, the rotation errors smoothly decreased as the data patch became smaller, (as this reduced the allowable range of the surface normal), but were worse than under assumption A3. The translation errors decreased - as the smaller patch sizes meant a smaller range of allowable positions, given the overlap assumptions.

When the experiments were run for the cone-cone case, the experimental setup was similar to the cylinder-cylinder case. The linear length and angular width were reduced by fixed percentages and the position estimated. The error tolerances were:

Model Value		Data Value		Data Uncertainity	
l_m	100	l_d	$100p$	δ_l	1.0
θ_m	1.0	θ_d	$1.0p$	σ	0.1
ρ_m	0.333	not used	not used	not used	not used
κ_m	0.03	κ_d	0.03	α	5.0
$\vec{n}_m$	(0,0,-1)	$\vec{n}_d$	transformed	ϵ	0.1
$\vec{a}_m$	(1,0,0)	$\vec{a}_d$	transformed	β	0.1
$\vec{p}_m$	(0,0,0)	$\vec{p}_d$	transformed	γ	1.0

The same statistics were obtained as for the plane-plane case and the experiments were run for both assumptions A2 and A3. The results for the experiments are shown in Tables 5 and 6.

Under assumption A3, the surface normal was progressively less constrained, whereas the axis was always estimated well and produced results comparable to the cylinder-cylinder case shown in Table 3. Since the normal was always partly constrained, this meant that there was no catastrophic breakdown in rotation estimation as the patch became small. Because the cone tip could be inferred from the normal, axis and observed curvature, the translation estimates were independent of patch size. The tip position error was moderate (largely because of errors in the observed curvature).

Table 5: Cone-Cone Parameter Estimation Statistics for Assumption A3

p (%)	Constr.	Angle	Quat. Width	Trans.	Trans. Width
100	100	0.31	0.75	11.11	53.83
90	100	0.41	0.88	11.11	53.83
80	100	0.45	0.92	11.11	53.83
70	100	0.52	0.99	11.11	53.83
60	100	0.59	1.03	11.11	53.83
50	100	0.61	1.06	11.11	53.83
40	100	0.64	1.10	11.11	53.83
30	100	0.66	1.14	11.11	53.83
20	100	0.70	1.18	11.11	53.83
10	100	0.75	1.21	11.11	53.83

Table 6: Cone-Cone Parameter Estimation Statistics for Assumption A2

p (%)	Constr.	Angle	Quat. Width	Trans.	Trans. Width
100	100	1.07	1.44	11.11	53.83
90	100	1.06	1.43	11.11	53.83
80	100	1.02	1.41	11.11	53.83
70	100	1.01	1.40	11.11	53.83
60	100	0.99	1.39	11.11	53.83
50	100	0.97	1.37	11.11	53.83
40	100	0.94	1.35	11.11	53.83
30	100	0.91	1.32	11.11	53.83
20	100	0.87	1.30	11.11	53.83
10	100	0.82	1.26	11.11	53.83

Under assumption A2, the surface normal was still always constrained, though not as well as under assumption A3. Here, the rotation errors smoothly decreased as the data patch decreased, but were worse than under assumption A3. Results were comparable to the cylinder-cylinder case errors shown in Table 4. As under assumption A3, the translation errors were constant.

When the experiments were run for the generic-generic case, the experimental setup was similar to the plane-plane case, except in one respect: a second curvature axis was estimated, and this allowed the introduction of two additional TV2 rotation constraint modules into the network (altogether pairing: $\vec{n}_d$ & $\vec{a}_{1d}$, $\vec{n}_d$ & $\vec{a}_{2d}$ and $\vec{a}_{1d}$ & $\vec{a}_{2d}$).

As before, the angular widths were reduced by fixed percentages and the position estimated. The error tolerances were:

Model Value		Data Value		Data Uncertainity	
θ_{1m}	1.0	θ_{1d}	1.0p	σ_1	0.1
θ_{2m}	1.0	θ_{2d}	1.0p	σ_2	0.1
κ_{1m}	0.05	not used	not used	not used	not used
κ_{2m}	0.10	not used	not used	not used	not used
$\vec{n}_m$	(0,0,-1)	$\vec{n}_d$	transformed	ϵ	0.1
$\vec{a}_{1m}$	(1,0,0)	$\vec{a}_{1d}$	transformed	β_1	0.1
$\vec{a}_{2m}$	(0,1,0)	$\vec{a}_{2d}$	transformed	β_2	0.1
$\vec{p}_m$	(0,0,0)	$\vec{p}_d$	transformed	γ	1.0

The same statistics were obtained as for the plane-plane case and the experiments were run for both assumptions A2 and A3. The results for the experiments are shown in Tables 7 and 8.

As with the cylinder-cylinder case under assumption A3, the surface normal was progressively less constrained, whereas the axis was always estimated well. The third axis constraint did not produce better rotation estimates than with the cylinder-cylinder case, because here the axis correspondences could not be made so accurately. The translation estimates also increased smoothly, but were better than those of the cylinder case, largely because of the different radii and the curved surface allowed less spatial separation than

Table 7: Generic-Generic Parameter Estimation Statistics for Assumption A3

p (%)	Constr.	Angle	Quat. Width	Trans.	Trans. Width
100	100	0.40	0.80	0.99	5.29
90	100	0.48	0.91	1.01	6.98
80	100	0.54	1.00	1.06	8.69
70	100	0.64	1.13	1.12	10.43
60	100	0.67	1.21	1.24	12.18
50	100	0.75	1.30	1.39	13.96
40	100	0.84	1.38	1.60	15.74
30	100	0.94	1.45	1.85	17.53
20	100	0.98	1.48	2.16	19.33
10	100	1.05	1.52	2.50	21.12

Table 8: Generic-Generic Parameter Estimation Statistics for Assumption A2

p (%)	Constr.	Angle	Quat. Width	Trans.	Trans. Width
100	100	1.13	1.64	8.37	39.29
90	100	1.13	1.63	7.71	37.83
80	100	1.13	1.63	7.08	36.32
70	100	1.13	1.62	6.47	34.77
60	100	1.13	1.62	5.87	33.18
50	100	1.13	1.61	5.30	31.54
40	100	1.13	1.61	4.76	29.86
30	100	1.13	1.60	4.25	28.16
20	100	1.12	1.59	3.76	26.42
10	100	1.11	1.58	3.31	24.67

along the cylinder.

Under assumption A2, the errors again smoothly decreased as the patch size decreased, but were worse than under assumption A3, as was expected. The rotation mean error and range were roughly constant, whereas the translation error and range decreased substantially as patch size decreased.

7 Examples Of Use

This section shows some examples of the use of the constraints in the context of the IMAGINE II system [10], currently under development. The main input for this system is

Undergraduate Vision Laboratory
Department of Artificial Intelligence
University of Edinburgh

Sent by helium
Wed Nov 28 20:29:22 1990
File: unspecified, frame 1 of 1

FIG 11

Cosine Shaded View of block from above

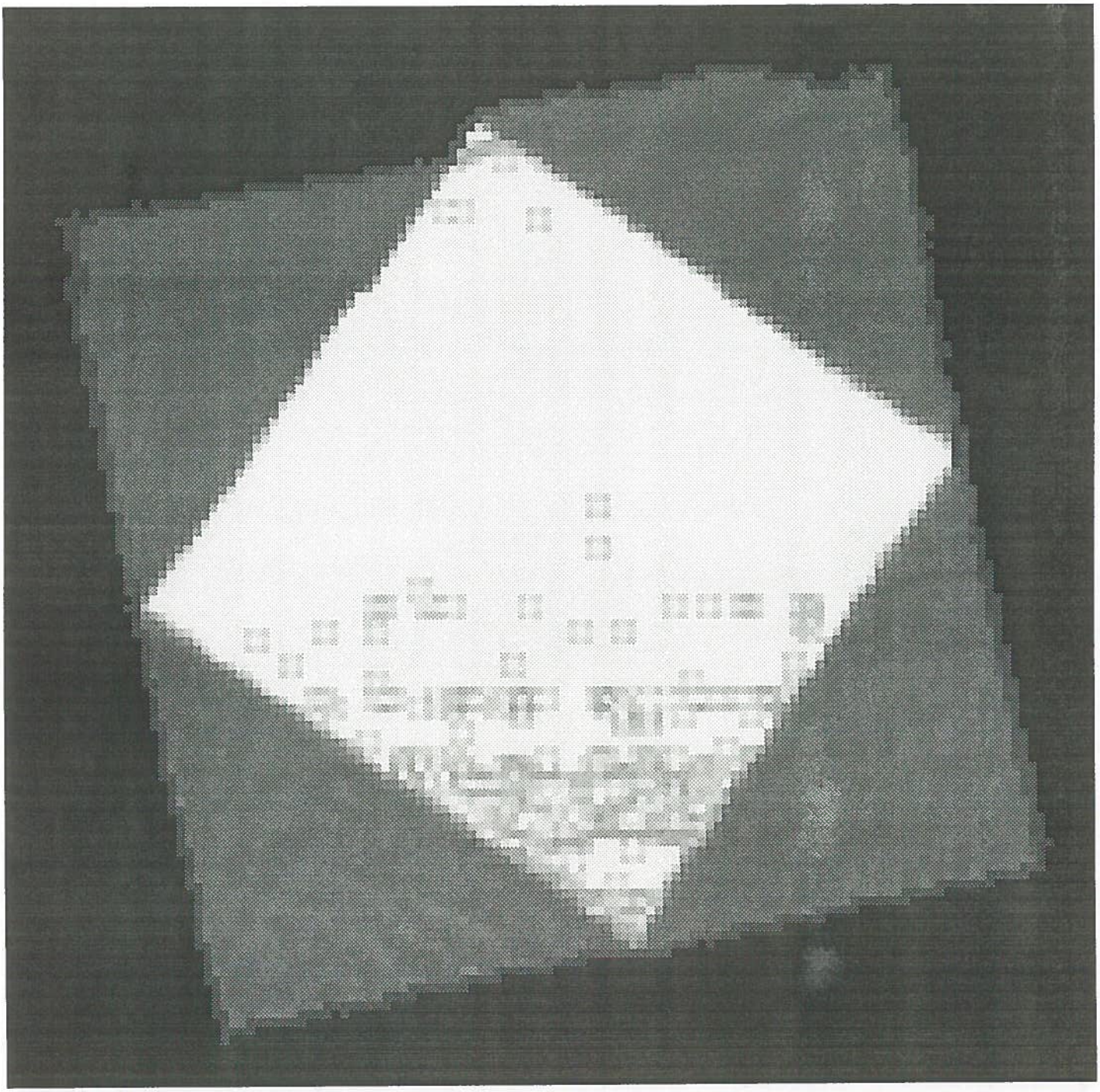


Figure 11: Cosine Shaded Range Data Of Test Part

dense range data, such as that for the block shown in Figure 11.

This data is segmented into a set of surface patches based on the signs of the principal curvatures. The main steps of the process are:

1. Detection of depth discontinuities,
2. Smoothing (but not across the depth discontinuities),
3. Detection of orientation discontinuities,
4. Calculation of mean (H) and Gaussian (K) curvatures,
5. Curvature sign-based classification of local surface shape,
6. Erosion and expansion to remove small patches and connect fragmented features and
7. Extraction of patches based on continuity and shape classification.

FIG 12

Light Grey Pixels ave segmented data patches

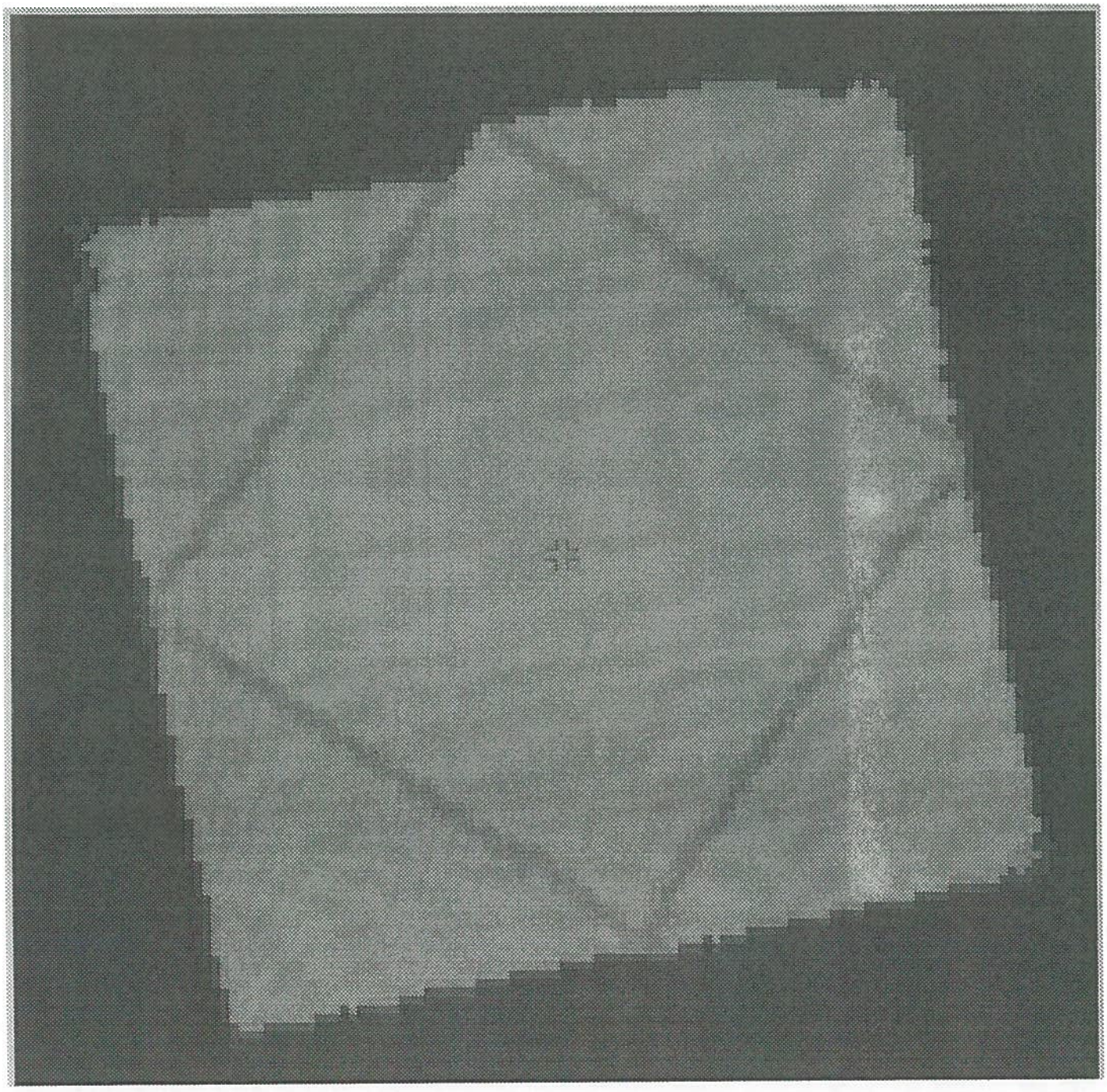


Figure 12: Surface Patches Extracted From Object

Figure 12 shows the surface patches extracted for the data shown in Figure 11

Descriptions of the surface patches are then produced, including:

- Surface areas and curvatures,
- Center-of-mass and surface normal at that point
- Boundary of patch
- Relative sizes and orientation
- Adjacency

The main use of the descriptions is in the model selection and position calculation processes.

The object models are defined using the SMS modeling language [9]. A synthesized image of the model of the block whose data was shown in Figure 11 is seen in Figure 13.

Undergraduate Vision Laboratory
Department of Artificial Intelligence
University of Edinburgh

Sent by helium
Wed Nov 28 20:13:50 1990
File: unspecified, frame 1 of 1

Fig 13

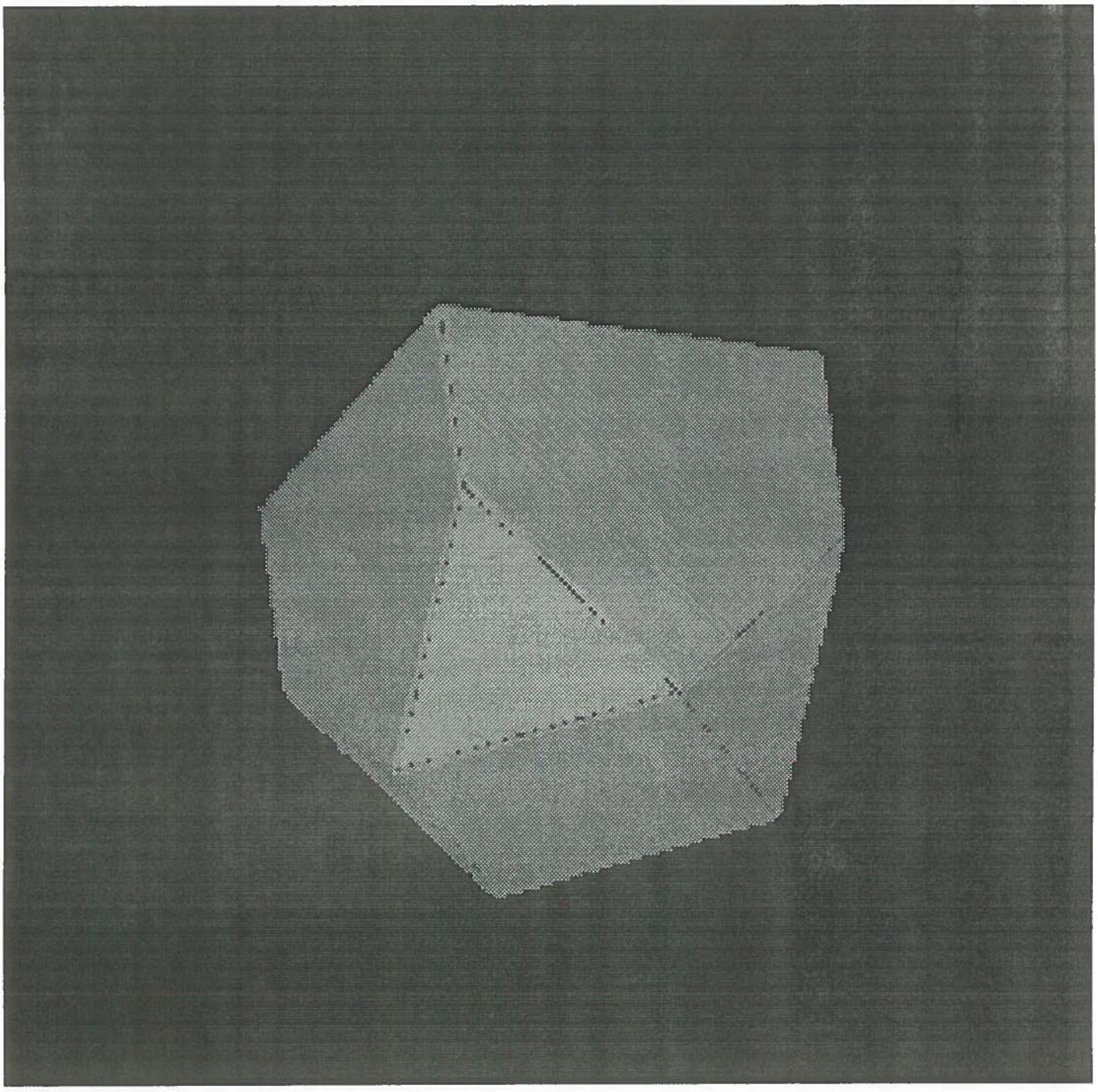


Figure 13: SMS Model Of Object

Model selection [11],[13] is based on an evidence accumulation process, where evidence comes from: (1) the degree to which data descriptions satisfy expected properties, (2) component, class and generic relationships, (3) general associations and (4) competitive inhibition. For this scene, the surface model patches selected for each data patch was the correct model only.

These model invocations drive the model-matching and position estimation process. The model-matching process is still under development, but works with polyhedral objects. It is at this point that the position constraints are generated, according to the theory developed in this paper (only some of the constraints have been implemented so far and some simplifications are used). These constraints are then input to the network-based geometric reasoning process to resolve the constraints.

For this scene, the extracted properties of the center data patch were:

$\vec{p}_d$ = (-1.0, 3.0, 900.5)

$\vec{n}_d$ = (-0.0017, 0.0092, -0.9999)

$\vec{a}_d$ = (0.1861, -0.9825, -0.001)

$\vec{c}_d$ = (-0.9825, -0.1861, 0.0034)

and the model properties were:

$\vec{p}_m$ = (35.4, 35.4, 0)

$\vec{n}_m$ = (0, 0, -1)

$\vec{a}_m$ = (0.7071, 0.7071, 0)

$\vec{c}_m$ = (0.7071, -0.7071, 0)

Figure 14 shows the graphs of $\omega_m(\phi)$ and $\omega_d(\phi)$. From this, we extract $\phi_d = 0$ (arbitrary choice), $\phi_m = 0.78$ and $\Omega = \{-0.87 \cdots -0.53\}$. (Note that there are four possible rotations compatible with the four-way rotational symmetry and this is just one of them).

Other parameters are:

FIG 14a

MODEL PATCH WIDTHS

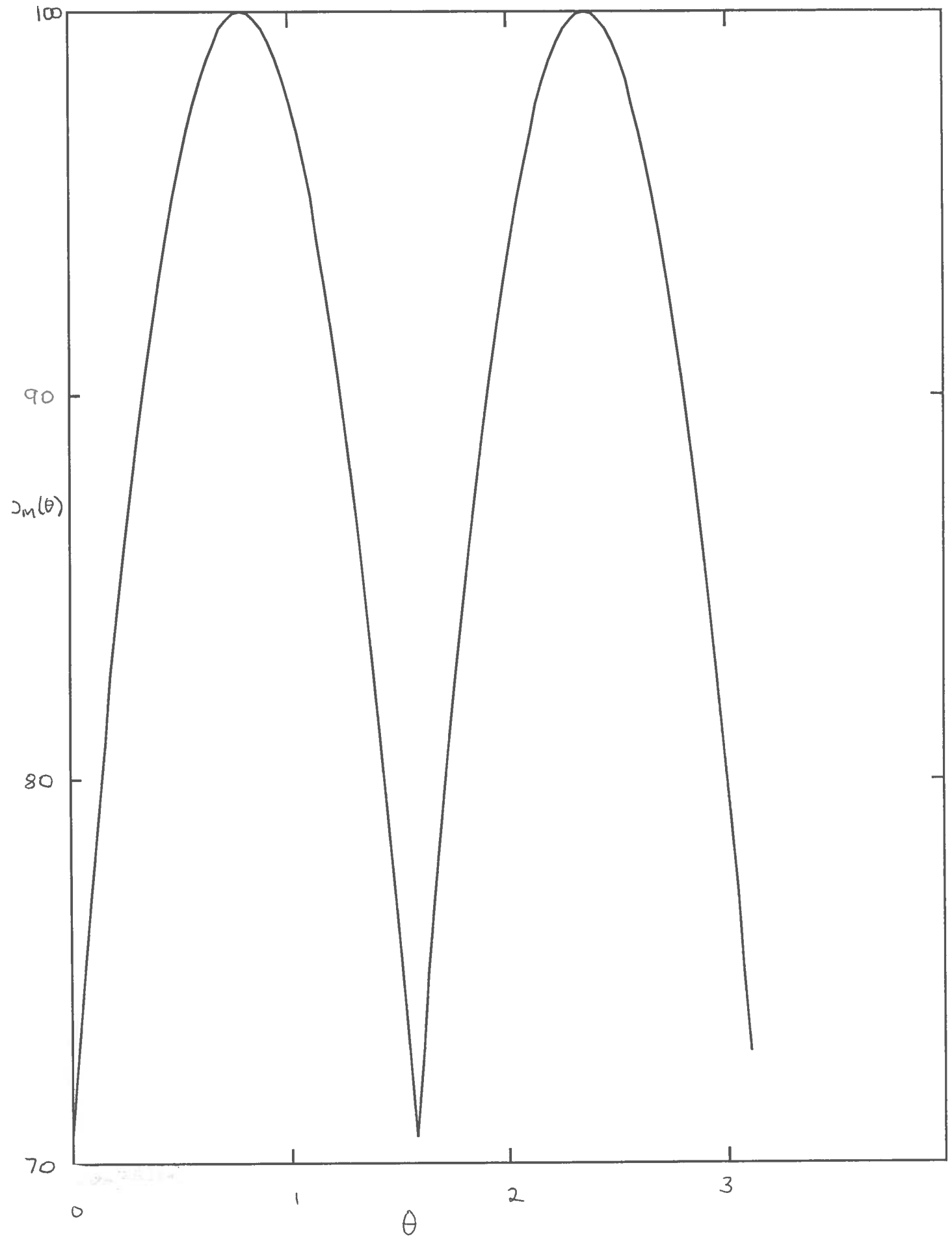


FIG 14b
Data Widths

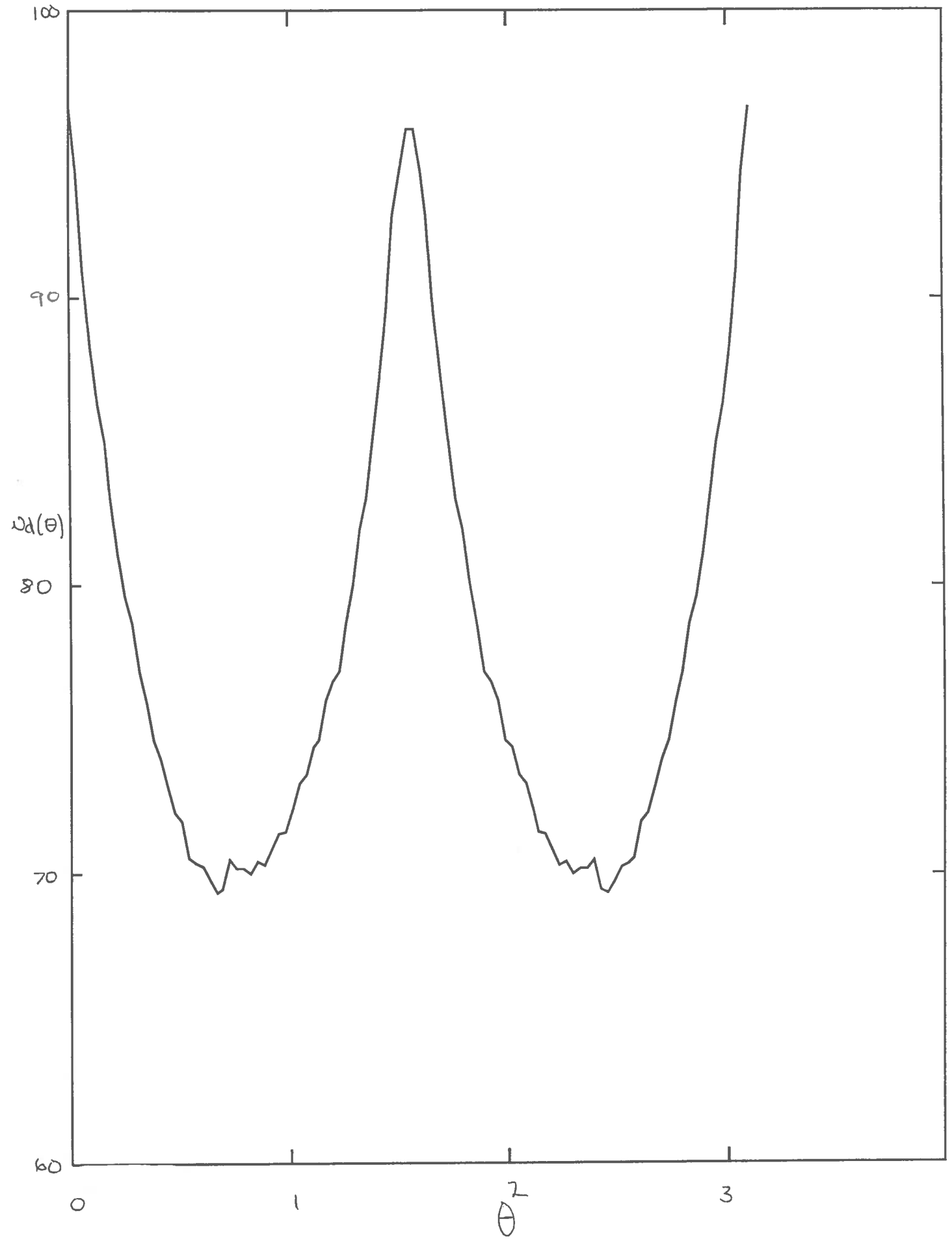


Figure 14: ω *versus* ϕ For Model And Data

$$
\begin{aligned}
\epsilon &= 0.02 \\
\gamma &= 1.0 \\
\beta &= 0.02 \\
U &= 100 \\
V &= 100 \\
\omega_d(\phi_d) &= 96.7 \\
\omega_d(\phi_d + \tfrac{\pi}{2}) &= 95.9 \\
A &= 0.07 \\
B &= 2.2 \\
C &= 2.6
\end{aligned}
$$

These values were used to derive the constraints described in Section 4.1 and were input into the network shown in Figure 10. Using this network, the evaluated position was:

```
Rotation quaternion:
        High: (0.553,0.058,0.066,-0.828)
```

Undergraduate Vision Laboratory
Department of Artificial Intelligence
University of Edinburgh

F16-15

Sent by helium
Wed Nov 28 21:31:43 1990
File: unspecified, frame 1 of 1

Cosine Shaded View of Block with Patch
Superimposed in Estimated Position

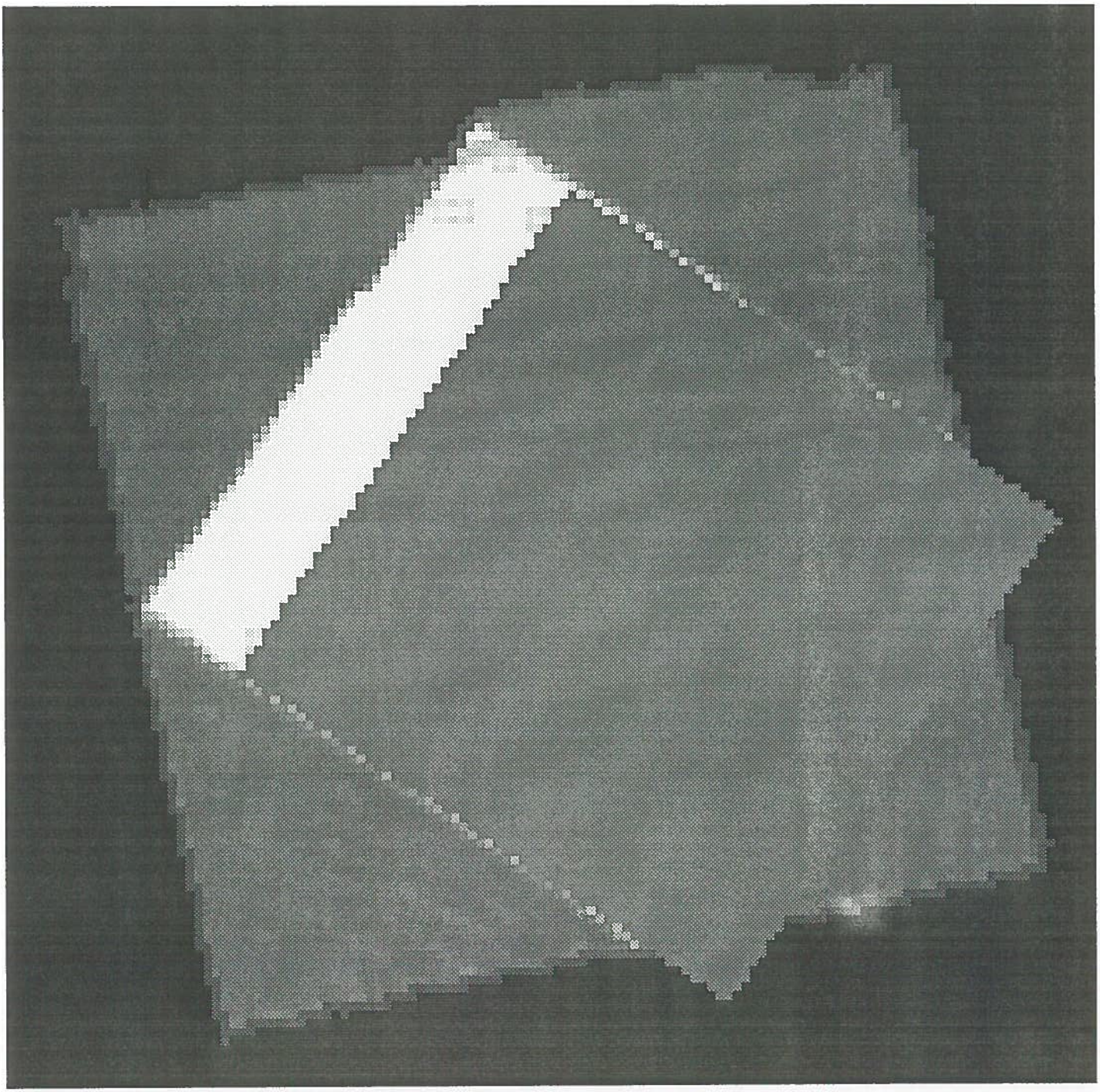


Figure 15: Estimated Position Of The Top Patch

```
        Low: (0.374,-0.070,-0.072,-0.927)
        Mean: (0.464,-0.006,-0.003,-0.878)
Translation vector:
        High: (16.233,58.130,912.048)
        Low: (-12.757,28.611,888.020)
        Mean: (1.738,43.370,900.034)
```

Using the mean estimated rotation and translation, a synthesized image of the model patch is seen in Figure 15, where it is drawn on top of the range data. There is some drift from the true position, but this is a position compatible with the position constraints given above. (It must be said that all positions compatible with the constraints are generated, but not all positions in the range of estimates satisfy all constraints, because this method of solving the constraints is sub-optimal.)

These position estimates were based on a limited set of constraints obtainable from a single patch. When multiple patches and constraints derivable from pairs of patches

Undergraduate Vision Laboratory
Department of Artificial Intelligence
University of Edinburgh

Sent by helium
Wed Nov 28 21:25:44 1990
File: unspecified, frame 1 of 1

FIG 1C

Block Model Superimposed on Range Data
(good overlap)

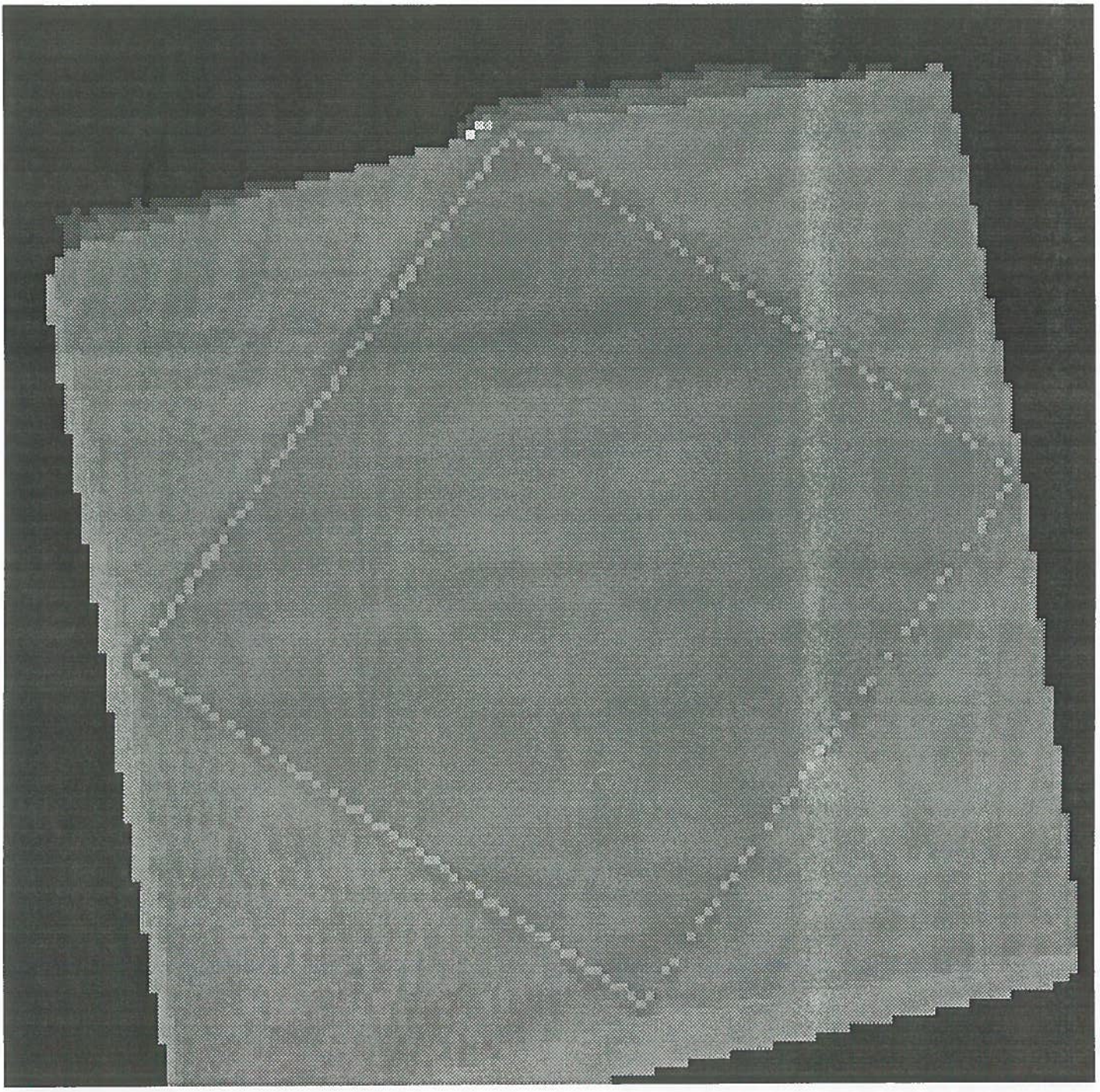


Figure 16: Estimated Position Of Whole Object

are used (e.g. [15]),better results occur. Figure 16 shows the model for the whole block superimposed on the raw data. Note that alignment is good enough that only a few data pixels are seen around the edge of the projected model. While the correspondence is quite good, the only rotation constraints used here were those on the patch normals.

We are still implementing the model matching for curved surfaces, but Figure 17 shows (a) cosine shaded range data for a head and (b) the superposition of a face model on the data, where the position of the face was estimated by the network technique using only surface normals and central points (patch correspondences were determined by hand).

8 Discussion

The paper identified several geometric constraints for estimating surface positions, given information about the shapes of the patches and the relationships between the model and data patches. The constraints were specified for what we felt were the most important patch shapes and model-to-data relationships. The experiments of Section 6 showed both that the constraints are valid, and how well they perform and the demonstration of Section

Undergraduate Vision Laboratory
Department of Artificial Intelligence
University of Edinburgh

Sent by helium
Wed Nov 28 23:09:27 1990
File: unspecified, frame 1 of 1

Fig 17a

Cosine Shaded Face

Undergraduate Vision Laboratory
Department of Artificial Intelligence
University of Edinburgh

Sent by helium
Wed Nov 28 23:09:21 1990
File: unspecified, frame 1 of 1

FIG 17b

Cosine Shaded Face with Superimposed Face model

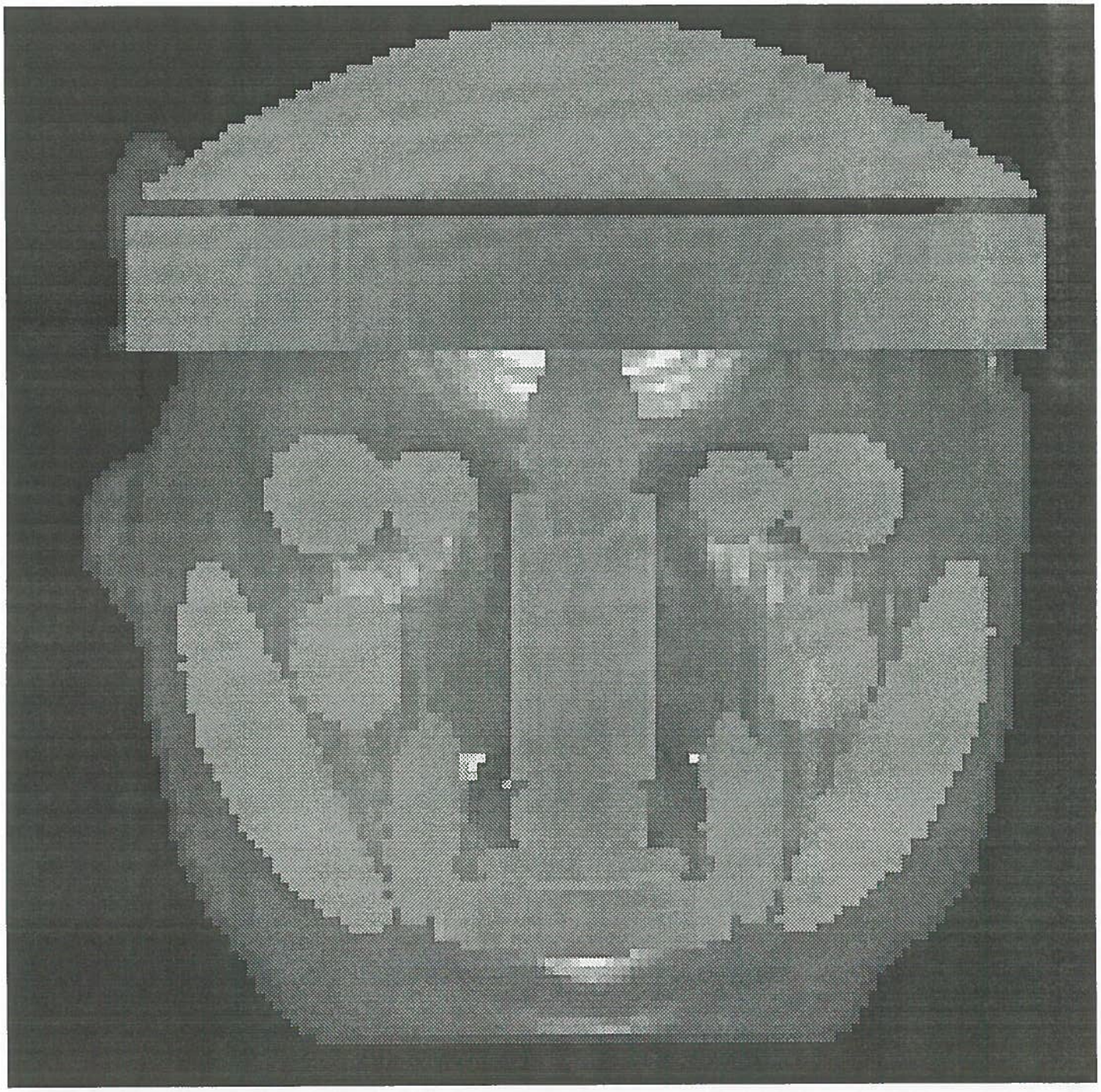


Figure 17: Estimated Position Of Face

7 showed how they can be used in a model-based vision system.

We would like to now discuss some of the implications of the constraints and experiments.

For simplicity and uniformity, we mainly used properties from a width bounding of the patches. Clearly, there are some boundary shapes (such as with a long, narrow protrusion) where this uniform bounding loses information. Other types of bounding could be developed; however the use of the width did allow a precise statement of the range of allowable rotations. Moreover, most of the patch types currently used in our scene analysis are fairly compact, so this bounding approach is appropriate.

We considered only four generic model-to-data patch relationships. Other relationships undoubtedly exist, especially in restricted scenes, and constraints similar to the above can be specified.

It is possible that tighter constraints could be developed for the cases. For example, the position constraints generally treated the three axis directions independently, and then transformed the bounds relative to these to bounds in the three coordinate direc-

tions. There is likely to be some dependence between the factors, allowing tighter position estimates.

The estimation of the patch orientation is not always accurate. This stems from three problems:

1. the axis constraints are not particularly strong, so the overall orientation is dramatically affected,

2. the TV2 rotation estimation module is not particularly effective when estimating rotations from inputs with substantial variation and

3. each stage of the TV2 rotation estimation process increases the size of the output intervals, so that modest errors are amplified in the output.

The result of these problems is that there are many rotation estimates where the precise direction of the rotation cannot be determined (i.e. determined to be $\vec{q}$ or $-\vec{q}$) and hence the output rotation intervals are large.

It is our belief that the position constraints from individual patches are generally weak. While some features are strong and reliable, such as the normals on planar surfaces or curvature axes on strongly curved surfaces, many other direction vectors are weak, and do not greatly constrain object position. Work in progress shows that using constraints from pairs of surfaces (e.g. two normals) gives better position estimates. For example, the vector between two distant, highly curved features gives much more reliable data than the surface normals on either. Adding a third vector also considerably strengthens the estimation process, because the results affect each other by value propagation.

Though the network-based constraint satisfaction process may not be ideal, the position constraints presented in this paper hold, irrespective of the process used to estimate the position parameters from the constraints.

Acknowledgements

This work was funded by the University of Edinburgh. I'd like to thank M. Cameron-Jones, J. Hallam, H. Hughes and M. Trucco for suggestions with the text and work.

Bibliography

1. G. Alefeld and J. Herzberger, Introduction to Interval Computations, Academic Press, London, 1983.

2. D. H. Ballard, H. Tanaka, "Transformational Form Perception in 3D: Constraints, Algorithms, Implementation", Proc. 9th Int. Joint Conference on Artificial Intelligence, pp 964-968, Los Angeles, 1985.

3. P. J. Besl, "Surfaces in Early Range Image Understanding", PhD Dissertation, Electrical Engineering and Computer Science Department (RSD-TR-10-86), University of Michigan, 1986.

4. R. A. Brooks, "Symbolic reasoning among 3-D models and 2-D images", Artificial Intelligence, Vol 17, pp. 285-348, 1981.

5. L. D. Cai, "Scale-Based Surface Understanding Using Diffusion Smoothing", Submitted PhD Thesis, Department of Artificial Intelligence, University of Edinburgh, 1990.

6. E. Davis, "Constraint Propagation with Interval Labels", Artificial Intelligence, Vol 32 (3), pp281-331, 1987.

7. H. F. Durrant-Whyte, "Uncertain geometry in robotics", IEEE J. on Robotics and Automation, vol 4(1), pp 23-31, 1988.

8. O. Faugeras, M. Hebert, "A 3-D Recognition and Positioning Algorithm Using Geometrical Matching Between Primitive Surfaces", Proc. 8th Int. Joint Conf. on Artificial Intelligence, Karlsruhe, pp 996-1002, 1983.

9. R. B. Fisher, "SMS: A Suggestive Modeling System For Object Recognition", Image and Vision Computing, Vol 5, No 2, pp. 98-104, 1987.

10. R. B. Fisher, "The Design of the IMAGINE II Scene Analysis Program", in Frisby and Mayhew (eds), 3D Model Recognition from Stereoscopic Cues, MIT Press, forthcom-

ing. See also Univ. of Edinburgh, Dept. of Artificial Intelligence, Working Paper 211, December 1987.

11. R. B. Fisher, From Surfaces to Objects: Computer Vision and Three Dimensional Scene Analysis, John Wiley and Sons Ltd, Chichester, 1989.

12. R. B. Fisher and M. J. L. Orr, "Solving Geometric Constraints in a Parallel Network", Image and Vision Computing, Vol 6, No 2, 1988.

13. R. B. Fisher, "Model Invocation for Three Dimensional Scene Understanding", Proc. 10th Int Joint Conf. on Artificial Intelligence, Milan, pp 805-807, 1987.

14. R. B. Fisher and M. J. L. Orr, "Experiments with a Network-Based Geometric Reasoning Engine", Proc. 11th Int Joint Conf. on Artificial Intelligence, Detroit, pp 1623-1628, 1989.

15. W. E. L. Grimson, T. Lozano-Perez, "Model-Based Recognition and Localization from Sparse Range or Tactile Data", International Journal of Robotics Research, Vol. 3, pp. 3-35, 1984.

16. R. M. Haralick, H. Joo, C. Lee, X. Zhuang, V. Vaidya, M. Kim, "Pose Estimation from Corresponding Point Data", in Freeman (ed), Machine Vision for Inspection and Measurement, Academic Press, pp 1-84, 1989.

17. R. Hoffman, A. Jain, "An Evidence-Based 3D Vision System For Range Images", Proceedings 1st IEEE International Conference on Computer Vision, pp. 521-525, 1987.

18. R. Horaud, "Combining Image and Spatial Reasoning for Model Retrieval", Proc. 7th European Conf. on Artificial Intelligence, Brighton, pp 529-538, 1986.

19. Z. Q. Li, "A Curve Analysis Approach to Surface Feature Extraction From Range Data", Proc. International Workshop on Machine Intelligence and Vision, Tokyo, April 1989.

20. D. H. Lowe, "Perceptual Organization and Visual Reconstruction", Kluwer Academic Publishers, Boston, 1985.

21. M. J. L. Orr and R. B. Fisher, "Geometric Reasoning for Computer Vision", Image and Vision Computing, Vol 5, No 3, pp 233-238, 1987.

22. Roberts, L. G., "Machine Perception of Three-Dimensional Solids", in J. T. Tippett, (ed), Optical and Electro-Optical Information Processing, MIT Press, Ch 9, Cambridge, Massachusetts, pp159-197, 1965.

23. E. Sacks, "Hierarchical Inequality Reasoning", TM 312, MIT Laboratory for Computer Science, 1987.

24. A. Zisserman, C. Marinos, D. A. Forsyth, J. L. Mundy, C. A. Rothwell, "Relative Motion and Pose from Invariants", Proc. 1st British Machine Vision Conference, Oxford, pp 7-12, 1990.

A Non-Planar Model Patches Paired with Planar Data Patches

A.1 Planar Data Patch - Model Cylinder

This case assumes that the observed data surface has a low curvature or has been tesselated. In any case, the scene analysis program has decided that the data surface should be paired with a cylindrical model feature. One characteristic of this case is that there is no curvature-based constraint on the orientation of the data axis about the complete cylinder. Another characteristic is that the origin of the model feature lies on its axis, rather than on its surface.

The only rotation constraint is given by the elongation of the patch, when compared to the axis of the cylinder. If A3 holds, we can relate the model and data elongation axes, within the observed angular error of β. This constraint limits the rotation of the model surface in the plane of the data surface, such that it still fits around the observed data patch:

Let:

$\omega_m(\phi)$ = boundary width (see Figure 2) of the flattened model surface at orientation ϕ

$\omega_d(\phi)$ = boundary width of the data surface at orientation ϕ

$\Omega = \{\theta : \forall\phi(\omega_m(\phi - \theta) \geq \omega_d(\phi) - \delta_\omega)\}$ is the set of possible rotations of the model that can fit around the data

$\vec{a}_m$ = the vector in the flattened model surface corresponding to the cylinder axis (at orientation ϕ_m)

$\vec{a}_d$ = average orientation of $\vec{a}_m$ over Ω (at orientation ϕ_d)

If $size(\Omega)/2 + \beta < \pi$ then:

$$T\vec{a}_m \cdot \vec{a}_d \geq cos(size(\Omega)/2 + \beta)$$

If more than one interval of angles exists in Ω, then each interval creates an independent alternative constraint.

The translation constraints have the form:

$$(T\vec{p}_m - \vec{p}_d + \frac{1}{\kappa_m}\vec{n}_d) \in I(A\vec{n}_d) + I(B\vec{a}_d) + I(C\vec{c}_d) + I(\vec{\gamma})$$

where the constants A, B and C depend on the specific assumptions as given below. Let:

$$M = min_\phi(\omega_m(\phi))$$

$$N = max_\phi(\omega_m(\phi))$$

$$U = max_\phi(\omega_m(\phi)) \text{ over } \phi_d - \phi \in \Omega$$

$$V = max_\phi(\omega_m(\phi + \pi/2)) \text{ over } \phi_d - \phi \in \Omega$$

Asmp.	A	B	C
A1	$\epsilon\sqrt{B^2+C^2}$	$R-\frac{1}{2}(M+\omega_d(\phi_d))-\delta_\omega)$	$R-\frac{1}{2}(M+\omega_d(\phi_d+\pi/2)-\delta_\omega)$
A2	$\epsilon\sqrt{B^2+C^2}$	$\frac{1}{2}(N+\omega_d(\phi_d)+\delta_\omega)$	$\frac{1}{2}(N+\omega_d(\phi_d+\pi/2)+\delta_\omega)$
A3	$\epsilon\sqrt{B^2+C^2}$	$\frac{1}{2}(U-\omega_d(\phi_d)+\delta_\omega)$	$\frac{1}{2}(V-\omega_d(\phi_d+\pi/2)+\delta_\omega)$

A.2 Planar Data Patch - Cylindrical Model Patch

This case exploits the plane-plane case of Section 4.1, but uses some substitutions from the cylindrical model features. The assumption is that the data is cylindrical, but noise has caused it to appear to be planar. This means that the curvature axis is not inferred, so only the boundary constraints can be used. The following substitutions are used:

Model Cylindrical Patch Feature	Corresponding Plane Model Feature
$\vec{a_{pm}}$	$\vec{a}_m$
$\omega_m(\phi)$	$\omega_m(\phi)$

A.3 Planar Data Patch - Model Cone

This case exploits the plane-plane case of Section 4.1, only using some substitutions from the conical model features. The assumption is that the data is conical, but noise has caused it to appear to be planar. This means that the curvature axis is not inferred, so only the boundary constraints can be used. The following substitutions are used:

Model Cone Feature	Corresponding Plane Model Feature
$\vec{a_{pm}}$	$\vec{a}_m$
$\omega_m(\phi)$	$\omega_m(\phi)$

A.4 Planar Data Patch - Generic Model Patch

This case exploits the plane-plane case of Section 4.1, only using some substitutions from the generic model patch features. The assumption is that the data is generic, but noise

has caused it to appear to be planar. This means that the curvature axes are not inferred, so only the boundary constraints can be used. The following substitutions are used:

Model Generic Feature	Corresponding Plane Model Feature
$\vec{a_{pm}}$	$\vec{a}_m$
$\omega_m(\phi)$	$\omega_m(\phi)$

B Non-Cylindrical Model Patches Paired with Cylindrical Data Patches

B.1 Cylindrical Data Patch - Planar Model Patch

This case exploits the plane-plane case of Section 4.1, assuming that the observed data cylindrical patch is a noise-corrupted planar patch. The data observations taken for the cylindrical patch are not directly usable for the planar patch, so the flattened boundary is used instead. The following substitutions are used:

Data Cylindrical Patch Feature	Corresponding Plane Data Feature
$\vec{a}_{pd}$	$\vec{a}_d$
$\omega_d(\phi)$	$\omega_d(\phi)$
δ_ω	δ_l
β_ω	β

B.2 Cylindrical Data Patch - Model Cone

If we could not estimate the cone's rate of radius change, we could use the reduced constraints by treating it as if it were a cylinder. The following substitutions are needed for using the constraints of Section 4.2.1:

Model Cone Feature	Corresponding Cylinder Feature
$(\kappa_{maxm} + \kappa_{minm})/2$	κ_m
$\vec{a}_m$	$\vec{a}_m$
θ_m	θ_m
l_m	l_m

B.3 Cylindrical Data Patch - Generic Model Patch

If we could not estimate the generic patch's second curvature magnitude, we could use the reduced constraints by treating it as if it were a cylinder. The following substitutions are needed for using the constraints of Section 4.2.1. Here there are two possible substitutions for the two axes.

Model Generic Feature ($i = 1$ or 2)	Corresponding Cylinder Feature
κ_{im}	κ_m
$\vec{a}_{im}$	$\vec{a}_m$
θ_{im}	θ_m
$w_{(3-i)m}$	l_m

C Non-Conical Model Patches Paired with Conical Data Patches

C.1 Conical Data Patch - Planar Model Patch

This case exploits the plane-plane case of Section 4.1, assuming that the observed conical data patch is a noise-corrupted planar patch. The data observations taken for the conical patch are not directly usable for the planar patch, so the flattened boundary is used instead. The following substitutions are used:

Data Cone Feature	Corresponding Plane Data Feature
$\vec{a}_{pd}$	$\vec{a}_d$
$\omega_d(\phi)$	$\omega_d(\phi)$
δ_ω	δ_l
β_ω	β

C.2 Conical Data Patch - Model Cylinder

We assume that noise has caused a portion of a cylinder to be observed as conical patch. The following substitutions are used in the constraints of Section 4.2.2:

Data Cone Feature	Corresponding Cylindrical Patch Feature
$\vec{a}_d$	$\vec{a}_d$
l_d	l_d
β	β
δ_l	δ_l

C.3 Conical Data Patch - Cylindrical Model Patch

We assume that noise has caused a cylindrical patch to be observed as conical. The following substitutions are needed for using the constraints of Section 4.2.1:

Data Cone Feature	Corresponding Cylindrical Patch Feature
$\vec{a}_d$	$\vec{a}_d$
θ_d	θ_d
l_d	l_d
σ	σ
β	β
δ_l	δ_l

C.4 Conical Data Patch - Generic Model Patch

We assume that noise has caused a generic patch to be observed as conical. Because we cannot align the features properly, we treat this case as if it were a cylinder-cylinder case. The two curvatures of the generic patch mean that two possible substitutions are allowed, but one might be eliminated because of incompatibility of curvature signs or magnitudes. The following substitutions are needed for using the constraints of Section 4.2.1:

Model Generic Feature ($i = 1$ or 2)	Corresponding Cylindrical Feature
κ_{im}	κ_m
$\vec{a}_{im}$	$\vec{a}_m$
θ_{im}	θ_m
$w_{(3-i)m}$	l_m

Data Cone Feature	Corresponding Cylindrical Feature
$\vec{a}_d$	$\vec{a}_d$
θ_d	θ_d
l_d	l_d
δ_l	δ_l
β	β
σ	σ

D Non-Generic Model Patches Paired with Generic Data Patches

D.1 Generic Data Patch - Planar Model Patch

This case exploits the plane-plane case of Section 4.1, assuming that the observed generic data patch is a noise-corrupted planar patch. The data observations taken for the generic patch are not directly usable for the planar patch, so the flattened boundary is used instead. The following substitutions are used:

Data Generic Feature	Corresponding Plane Data Feature
$\vec{a}_{pd}$	$\vec{a}_d$
$\omega_d(\phi)$	$\omega_d(\phi)$
δ_ω	δ_l
β_ω	β

D.2 Generic Data Patch - Model Cylinder

Perhaps a cylinder was observed as a generic patch because of noise, but we use constraints derived from treating it as if it were a cylinder. The following substitutions are needed for using the constraints of Section 4.2.2. There are two possible substitutions for the two generic patch axes.

Data Generic Feature ($i = 1$ or 2)	Corresponding Cylinder Feature
$\vec{a}_{id}$	$\vec{a}_d$
$w_{(3-i)d}$	l_d
β_i	β
δ_i	δ_l

D.3 Generic Data Patch - Cylindrical Model Patch

Here, a cylindrical patch was observed as a generic patch because of noise, but we use constraints derived from treating it as if it were a cylinder. The following substitutions are needed for using the constraints of Section 4.2.1. There are two possible substitutions for the two generic patch axes.

Data Generic Feature ($i = 1$ or 2)	Corresponding Cylinder Feature
$\vec{a}_{id}$	$\vec{a}_d$
θ_{id}	θ_d
$w_{(3-i)d}$	l_d
β_i	β
δ_i	δ_l
σ_i	σ

D.4 Generic Data Patch - Model Cone

We assume that noise has caused a conical patch to be observed as generic. Because we cannot align the features properly, we treat this case as if it were a cylinder-cylinder case. The two curvatures of the generic patch mean that two possible substitutions are allowed, but one might be eliminated because of incompatibility of curvature signs or magnitudes. The following substitutions are needed for using the constraints of Section 4.2.1:

Data Generic Feature ($i = 1$ or 2)	Corresponding Cylindrical Feature
$\vec{a}_{id}$	$\vec{a}_d$
θ_{id}	θ_d
$w_{(3-i)d}$	l_d
σ_i	σ
β_i	β
δ_i	δ_l

Model Cone Feature	Corresponding Cylindrical Feature
$\frac{1}{2}(\kappa_{maxm} + \kappa_{minm})$	κ_m
$\vec{a}_m$	$\vec{a}_m$
θ_m	θ_m
l_m	l_m