Matching 'sticks, plates and blobs' objects using geometric and relational constraints

Prasanna G Mulgaonkar, Linda G Shapiro and Robert M Haralick

A recognition scheme using relational and rough geometric information about three-dimensional man-made objects to recognize instances of the objects in single perspective-normal views of scenes is described. Experiments performed using the matching scheme show that, in most cases, the object in the view can be identified correctly and reasonable estimates can be made of the unknown camera position responsible for generating the given view.

The technique is based on the fact that the camera position constrains the appearance of the various parts of the object. The propagation of these constraints from one planar object surface to another through the projection equations is worked out. This constraint propagation guides the matching scheme in the development of the interpretation of the scene. The results provide an estimate of the camera position within 20° of the actual location.

Keywords: perspective projections, relational matching, constraint analysis

Given an image of a scene containing three-dimensional objects, we should like a computer vision system to be able to name or describe the objects in the scene. The performance of this task by a computer is still a major research problem, although it can be trivially performed by human beings. Humans use past experience in seeing and touching the same or similar objects in the recognition process. It is now a well accepted fact that the computer must also employ knowledge to perform recognition tasks.

Human beings also exhibit the remarkable ability to recognize common objects from crude and incomplete descriptions. For example, most everyday objects can be identified from silhouettes. Even crude pictures drawn by children retain enough information to permit a guess as to the depicted object. This seems to suggest that the knowledge base of humans consists of descriptions which can tolerate the loss of considerable amounts of information. The recognition or matching processes which identify and classify the sensory input also have to be forgiving and able to perform inexact matching operations.

The main goal of this research was to produce such a rough knowledge base on a computer and to design a simple matching scheme, which would still be able to carry out recognition tasks with a certain degree of success. The rough models and simple matching scheme would then be used at the top level of matching in a computer vision system.

Three-dimensional object models are one form of knowledge that can be given to a computer program. Most of the current modelling techniques build descriptions of objects from simpler primitives. We have chosen to use rough relational models (the 'sticks, plates and blobs' models[1]) for the first step in the matching process. Our models consist of a global property list, the properties of each primitive, binary connections and related angles, ternary connections and related angles, perpendicular and parallel pairs, and additional constraints. Of these relations, only the unary, binary and ternary relations were used by the matching process. These models (described in the next section) are translation, rotation and scale independent.

Given a particular kind of object model, another form of knowledge involves the influence of the position of the camera viewing such an object on the appearance of the resultant image. In this paper we describe a recognition system that uses relational and geometric constraints developed especially for the sticks, plates and blobs models to recognize three-dimensional man-made objects from single perspective views. The next section gives a brief literature review. In the third section the relational models are defined in detail, including the geometric knowledge of objects represented by these models. The fourth and fifth sections describe the matching process.

LITERATURE REVIEW

We first present a brief review of some of the related work by other researchers. This is by no means a complete survey. It is mainly intended to be indicative of the variety of techniques that have been examined.

Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA

 0262–8856/84/02085–14$03.00

Object modelling

A large volume of work has been reported in the field of three-dimensional object modelling and representation. Most of the current techniques build up descriptions of objects from simpler primitives in various ways. Constructive solid-geometry[2] systems use set-theoretic 'additions' and 'subtractions' of solid primitives to assemble objects. Binford[3] first proposed a scheme of decomposing objects into 'generalized cylinders'. The generalized cylinder modelling was incorporated in a system for performing scene analysis experiments by Nevatia[4]. This technique was extended to a hierarchic system by Marr and Nishihara[5]. Generalized-cylinder models theoretically allow for very precise descriptions of object primitives, since both the axis functions and the sweeping functions may be arbitrarily specified. However, in practice, only very simple functions have been used so far. We selected the sticks, plates and blobs models because we needed only very rough descriptions of primitives at the first level of matching and because these models seem to correspond very well to the man-made objects we are working with.

The idea of organizing models and pictures into databases for scene analysis tasks has been around for a long time. Although organization of three-dimensional models into databases is not the main point of this paper, we include references to some recent papers for completeness. Interested readers may refer to the paper by Thomason and Gonzalez[6] for a treatment of database representation in scene analysis. The paper by Zhang Shouxuan[7] describes a pictorial database for recognition of Chinese characters. Organization of relational database of rapid access has been described by Shapiro and Haralick[8].

Shape decomposition

Decomposition of images into 'meaningful' parts has also received a considerable amount of attention. If an entire object of interest can be extracted from an image, then we should like to decompose this object into simpler parts before matching. Conceptually, the parts that are obtained should correspond to projections of the three-dimensional primitives used for modelling. Algorithms for two-dimensional shape decomposition have been reported by several researchers. A compilation of the major techniques has been given by Pavlidis[9]. The graph-theoretic clustering algorithm used in our work is based on the visibility of boundary points as seen from other points around the boundary. This algorithm is due to Shapiro and Haralick[10]. Approximation of the projections of three-dimensional objects by ellipses was reported by Gennery[11] for use in autonomous robot rover research. Gennery's camera solver program was also able to determine some of the parameters of the camera viewing the scene on the basis of two views. However, Gennery's three-dimensional primitives were rocks constrained to lie on a flat ground plane and two views were required to compute the camera parameters. In our work we use specific and more complex three-dimensional models and attempt the camera angle computations from a single view.

Matching schemes

Brooks[12, 13] used symbolic reasoning in the context of recognizing three-dimensional objects from single perspective views. His work differs from ours in that his constraints are encoded as symbolic expressions and are propagated from one part to another using symbolic manipulation techniques. Further, his models are considerably more exact than ours. Our propagation techniques use numerical calculations to predict the appearance of subparts of the objects and our models are, by design, very rough.

Relational matching of polygonal shapes has been reported by Shapiro[14]. Moravec's[15] robot cart used correlation techniques to match stereo views of the scene to generate a description of the three-dimensional world around it. Matching of single two-dimensional views with three-dimensional models has been reported by Barrow[16] in the context of aerial views and symbolic maps. The technique involved parametric correspondence and chamfer matching. However, in that research the models were not relational and required a good initial estimate of the camera position for the optimization technique to converge.

Computation of camera locations by determining correspondences between inexactly extracted landmarks in aerial views and their three-dimensional locations has been reported by Fischler and Bolles[17] using the Ransac technique. Marr and Nishihara[5] reported relaxation-based matching of three-dimensional stick-figure models to the two-dimensional projections of the axes of the generalized cylinder.

Relational matching traditionally has required large searches for which the time taken increases exponentially with search size. The use of discrete relaxation for the matching process was formalized by Rosenfeld *et al*[18]. Later Haralick and Elliott[19] examined speed-ups and tree pruning techniques that could be used for speeding up the tree searches used in such matching processes.

GENERALIZED BLOB MODELS

The modelling scheme used in our work is the generalized blob scheme developed by Shapiro *et al*[1]. This technique has been described in detail by Mulgaonkar[20]. In this paper we review the features required for the geometric reasoning processes.

Description of three-dimensional objects

The generalized blob model describes three-dimensional objects in a rough relational framework. The modelling scheme describes three-dimensional objects in terms of their constituent primitive parts. All models are decomposed into three basic shapes. These are sticks, plates and blobs and are shown in Figure 1. Sticks are inherently linear features, like chair legs, and are modelled as straight lines in three-space. Plates are the flat parts like chair backs or table tops. These are modelled as circular disks in three-space. Blobs are modelled as spheres and are the parts that occupy a large volume. An object contains a list of its parts, along with the geometric and relational interactions between them.

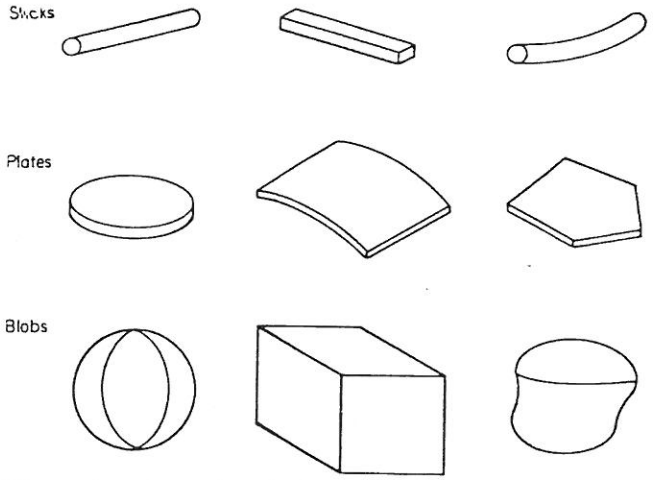

Figure 1. Examples of sticks, plates and blobs

In this work we use only three relations

- the 'simple parts' relation
- the binary 'connects/supports' relation
- the ternary 'triples' relation.

The angle information in the binary and ternary relations is also used in the geometric reasoning process. Since the models are supposed to be inexact, there is an implicit tolerance on the measurements specified in all relations.

We now give a formal definition of the sticks, plates and blobs data structure.

A stick is a 4-tuple

$$ST = (En, I, Cm, L)$$

where En is the set of two end points of the stick, I is the set of interior points of the stick, Cm is its centre of mass, and L is its length. Since straightline segments have each of the components of a stick, we shall be able to represent all sticks informally by straight-line segments to simplify our thinking about them.

A plate is a 4-tuple

$$PL = (Eg, S, Cm, A)$$

where Eg is the set of edge points; $S = \{S_1, S_2\}$ is the set of surface points of the plate, partitioned into the two surfaces; Cm is the centre of mass; and A is the area. Again, to simplify analyses, we can informally represent all plates by circles.

A blob is a triple

$$BL = (S, Cm, V)$$

where S is the set of surface points, Cm is the centre of mass, and V is the volume of the blob. We can informally represent all blobs as spheres.

We choose line segments, circles and spheres because they have no corners that we might be tempted to use in our descriptions. At the top level, the descriptions are to be as general and as rough as possible.

Geometry of a binary connection

In this section we examine the way in which a binary connection would be encoded in the database. In particular, we study the nature of a plate–plate connection in which the edges of the plates touch each other. This kind of contact occurs very often in man-made objects. For example, the back of a chair and the seat touch in this way (see Figure 5 below).

Connection of two plates

As was described in the previous section, plates are modelled as three-dimensional planar circles. Assume for the purposes of this illustration that both plates have unit radius. Let us examine how many geometric parameters have to be specified to describe the possible ways in which the two plates can touch.

Since the connection is described in terms of a local coordinate frame, the coordinate directions can be chosen to make the analysis as simple as possible. In particular, we choose the origin at the centre of one of the plates, say plate A (Figure 2), with the Z axis normal to the plane of A. Let the X axis lie along the line joining the centre of A to the point of its contact with plate B.

Since the radius of B is known, the centre of B can lie anywhere on a sphere centred at P (the point of contact between A and B). Imagine a polar coordinate system centred at P. The centre of B has two degrees of freedom. Therefore two angles are enough to describe the ray from P to the centre of B. The plane of plate B has one more degree of freedom, since this plane is only required to pass through the ray just fixed. Consequently three angles are enough to characterize the entire plate–plate edge–edge geometry.

The three angles actually measured are shown in Figure 2. Angle α measures the elevation of the centre of the second plate with respect to the first. β is the swing angle of the second plate. δ is the angle between the normals to the two plates. It should be pointed out at this stage that the encoding of the angles for the connection between parts A and B is not necessarily the same as the encoding for the connection between parts B and A. The reason for this is that the angles are specified in terms of coordinate systems centred at one of the parts (by convention, the first part). This is not necessarily a drawback since, given one encoding, it is possible to compute the other, resulting in conceptual simplicity at the expense of computational speed.

To achieve rotational and reflectional independence, the angles are constrained to lie within certain narrow ranges. The angle α can lie between 0° and 90°, β between 0° and 180°, and δ between −90° and +90°. Note that α and β do not have signs. Since there is no global coordinate system, there is no way of specifying clockwise or anticlockwise rotations. This is precisely what makes the description insensitive to mirror image reflections. However, because of this feature, up to eight

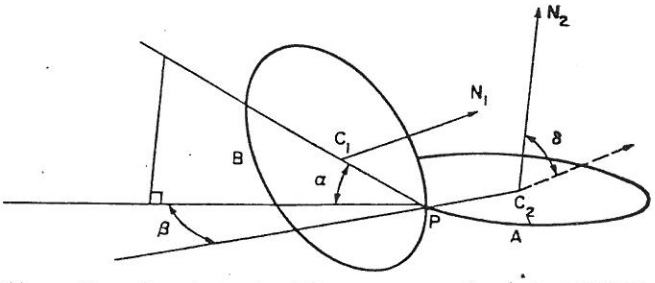

Figure 2. An edge–edge binary connection between two plates: C_1 and C_2 are the centres of the plates; $\mathbf{N}_1$ and $\mathbf{N}_2$ are the normals to the plates at their centres; α, β and δ are the angles tored in the connects/supports relation

physical interpretations can be constructed if the three angles are given. Figure 3 shows the four different orientations of the ray from P to the centre of B for given angles α and β. For each of these orientations there are two possible orientations for the angle δ. Figure 3 also shows how these eight descriptions are related; they are reflections about the XZ plane, which contains the normal to A and the line from the centre of A to P.

At the expense of possibly having to examine more than one interpretation for the three-dimensional object, we achieve the simplicity of having the description of a chair remain the same if the chair were viewed in a mirror.

Other primitive connections

Not all part pairs require three angles. Some connections, such as a stick–stick connection in which the ends of the primitives touch, require only a single angle. No connection type needs more than three angles in any case. For a complete explanation of all the angles necessary to describe every possible pair of primitives, see the paper by Mulgaonkar[20].

The three angles are used in the process of obtaining an interpretation of two-dimensional projections of objects. However, before we can examine the computations involved we need to look at the nature of the process by which the images of the scene are generated from physical three-dimensional objects.

Perspective projections and the camera geometry

The view that is generated from an object in the real world is the result of the interaction between the camera geometry and the surfaces and parts of the object.

Perspective normal projection

The projection of a point in three-dimensional space onto the camera screen is shown in Figure 4. The location of the point can be expressed in screen coordinates as

$$\frac{xs}{x'} = \frac{ys}{y'} = \frac{f}{d}$$

The camera itself is located with its origin at (X_c, Y_c, Z_c) in the world coordinate system. Without loss of generality

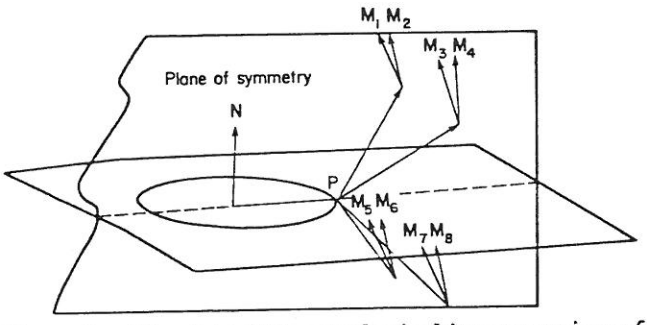

Figure 3. The eight different physical interpretations of a logical edge–edge description of two plates specified by three angles: each vector $\mathbf{M}_i$ indicates a different orientation of the second plate with respect to the first

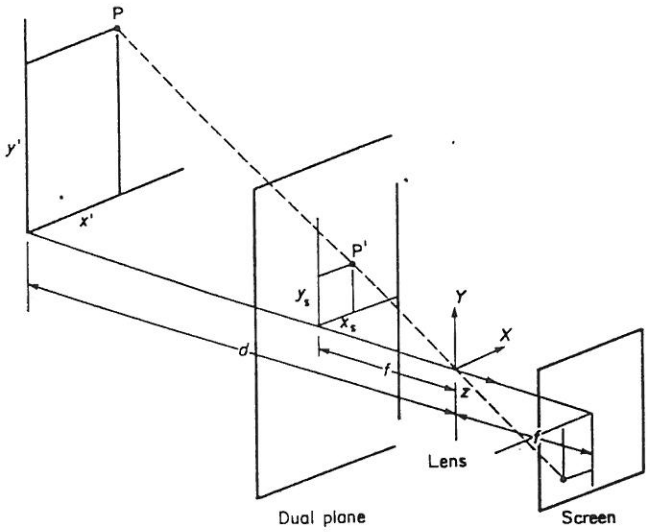

Figure 4. The projection of point P in three-dimensional space onto a camera screen behind the lens and the dual plane in front of the lens

we can assume that the negative Z axis of the camera points towards the origin in world coordinates. If it does not, it can be made to do so by a simple rotation of the appropriate coordinate frame.

Given the physical coordinates of the object parts, the focal ratio of the camera and the location of the camera in terms of the object coordinates, the exact image of the object can be mathematically generated. However, even if we are given the exact image of a three-dimensional object it is not possible to compute the inverse of the perspective transformation, since every point in the image is the projection of an entire line in three dimensions. This line is shown for an arbitrary point P in Figure 4.

The problem we are faced with is that we know even less than the information indicated above. We do not know the exact camera location (even though we may have some *a priori* information about the possible range of locations that the camera could occupy). For example, in aerial photography we can assume that the height of the camera above the ground plane is larger than the horizontal scale of the object.

To make the problem mathematically tractable we can make some simplifying assumptions about the nature of the perspective projection involved. In particular, we can assume that the swing angle is zero and that the Y axis of the camera coordinate system points in the same direction as the Z axis in the world. That is to say, all our views are 'right side up'.

Further, we can assume that the camera is located at a very large distance from the object and that the focal length of the lens is large. The resulting projection is called a perspective-normal projection (*cf* the work of Brooks[13]), because it is the equivalent of a normal projection onto a plane parallel to the screen and close to the object, followed by a perspective projection of that image onto the camera screen. This leaves just two unknowns necessary for specifying the camera location. These two parameters are the tilt and the pan angles of the camera as illustrated in Figure 5.

Projection of primitive parts

What do the three primitive parts of our objects look like

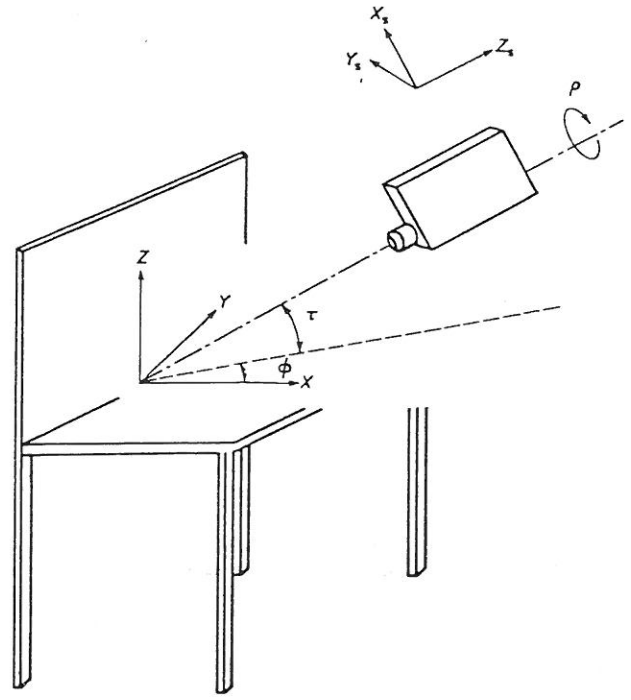

Figure 5. The tilt and the pan angles of the camera — the two unknown parameters of the perspective-normal projection (we assume that the roll angle of the camera is fixed)

under perspective-normal projection? Blobs have been described as spheres in three-space. The normal projection of a sphere onto any arbitrary plane is simply a circle. Sticks project as lines (or, depending on the viewpoint, they vanish). Plates are the most interesting since their projections yeield the most information about the relationship of the camera to the object.

Plates are modelled as circles. The normal projection of a three-dimensional circle is an ellipse. If the angle between the plane of the circle and the plane of the screen is θ (Figure 6), the eccentricity of the ellipse is sin $(90° - \theta)$. Perspective-normal projection is simply a normal projection with a constant scaling factor in both the X and Y directions and consequently does not change the eccentricity. Further, under our assumption that the camera is very far from the centre of our object, the scale factor is close to 1.0. Note that the angle θ in Figure 6 is 90.0° — the tilt angle shown in Figure 5. This means that the projection of a plate yields some information about the picture-taking process. The use of this information is the subject of other sections below.

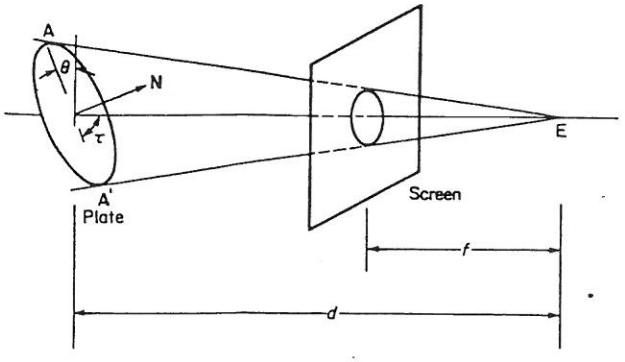

Figure 6. Perspective-normal projection of a plate (circle) in three-space

Projections of sticks also carry information about the relative spatial arrangement of the camera and object. Foreshortening of lines under projection can be used to extract information about the inclination of the line to the picture plane. in the work being reported in this paper, the information in projections of sticks was not used. Only the eccentricities of the projections of plates were considered.

In real objects, plates are not always exact circles. For example, the tops of tables (which would be modelled as plates) could be square or rectangular. This causes the observed eccentricity of the part to be less than the theoretically predicted value. The result of this discrepancy is discussed in later sections.

Estimation of camera parameters

In the previous subsection we showed that the appearance of a plate depends on its inclination with the viewing direction. If we have two plates which touch edge to edge, we can show that the tilt and pan angles of the viewing vector measured with respect to one of the plates depends on the tilt and pan angles with respect to the other plate. The camera position serves as a global constraint on the appearance of various parts of the object. If we know what one of the parts of the object looks like, its appearance provides some information about the possible range of camera positions. Propagating this information to adjacent parts yields a system by which these constraints can be verified and used in narrowing down the possible range of camera locations.

Since our models do not have any global coordinate system, we cannot specify the camera position in an absolute sense. We can, however, specify the tilt and pan angles of the camera with respect to local part-centred coordinates. In Appendix 1 we show that, once we know the tilt and pan angles with respect to one plate, we can propagate them over to adjacent plates which that plate touches, and consequently over all connected groups of plates in the model.

The calculations involved express the connection geometry of the plates in terms of vector equations which can be solved. Propagation of the constraints over all touching plates yields a global estimate of the camera position which relates the model to the view.

USING CONSTRAINT PROPAGATION IN A TREE SEARCH

From the last section we can see that the pan and tilt angles of the camera, specified with respect to any plate, effectively act as a constraint on the position of the camera. As we showed, this constraint can be propagated over to adjacent plates, and consequently over all the connected plates in the object. The measured tilt and pan angles have to meet the consistency checks described below.

Geometric constraints

As previously described, the geometric constraints arise from the fact that a single camera position generates the entire image. These constraints are imposed by requiring that, when the propagation of constraint angles yields a

value for the estimated tilt angle of any plate, the tilt angle must be compatible with the value computed from its eccentricity.

Relational constraints

The images generated from three-dimensional objects must also satisfy a set of relational constraints. If a set of primitives in the object participate in connection relations, and if all the parts are also visible in the view, their projections should also satisfy the equivalent two-dimensional connection relations. This is not a two-way implication. Parts that touch in the view do not necessarily correspond to connected three-dimensional primitives.

Putting it all together

The entire matching strategy can now be described. The process consists of finding a consistent interpretation for all the visible parts in the view. This necessitates a pre-processing phase in which the image is decomposed into constituent two-dimensional regions (possibly overlapping) which (intuitively) should correspond to the projections of the three-dimensional model parts.

Note that the main objective of the research described in this paper is the comparison of decomposed two-dimensional views with the rough three-dimensional model descriptions. The two decomposition schemes used were only for generating data for the matching processes. It is not our intention to present these schemes as the 'best' or 'only' applicable decomposition techniques. We hope that our low- and medium-level vision work[21] will lead to much better initial decomposition methods.

Two-dimensional preprocessing

Human beings have the ability to recognize fairly complex structures when presented with the silhouettes of such objects. This seems to indicate that in most cases the outer boundaries of man-made objects contain enough information to be able to characterize the object. One part of our experiments used such a two-dimensional boundary representation of projections of objects as input to the matching procedures. These outlines were decomposed into the constituent parts using a graph-theoretic clustering procedure[10] for generating near-convex clusters. The input is an ordered sampling of the points on the outer and inner (hole) boundaries of the silhouette of the projection of the three-dimensional object. Since this method does not retain any information about the lines internal to the object, it does not work well in all cases. However, for many models the outer boundary was found to retain enough information for a meaningful decomposition.

Some of the two-dimensional views that were used were obtained from digitized photographs of toy furniture. Other views were generated by computer from accurate three-dimensional descriptions of sample objects from known camera positions. One set of experiments was run using the input from the clustering algorithm. A second set of experiments involved ideal computer-generated decompositions in which all interior and exterior lines, along with the hidden lines, were used. The results of these experiments will be described later.

Matching process

Once a decomposition has been obtained, the next stage is to find a model to compare it with. At the simplest level, we can try each model in the database one at a time until a match is found. Although this may be feasible for small databases, the task may become prohibitively costly for large collections of models. One solution to this is to organize the database into clusters of similar models and to represent each cluster by a representative model. The description of the decomposition is first compared with the representatives and only those groups deemed similar enough are investigated further. The clustering approach to relational model database organization and a binary tree approach have been described by Shapiro and Haralick[8]. A related, but faster, tree approach was given by Feustel and Shapiro[22]. A simple approach to the problem, using bit vectors, was given by Nevatia and Binford[4].

Since the database used for testing purposes during this research was small, we performed a sequential search of all models without enforcing any structure on the database. As the model collection size increases, we shall organize the models according to one of the different retrieval schemes.

Once the model to be compared has been determined, comparison proceeds as follows. For every three-dimensional part in the model, we try to select a near-convex polygon in the view. As these assignments proceed, the tilt and pan angles computed at each stage refine the previous estimates for the camera position. The geometric consistency criterion is used to validate each possible instance by comparing the predicted tilt angle with the computed value. If the values do not lie in the predicted range, that association is ruled out. Note that tilt and pan angles are meaningful only in the case of plates. For sticks and blobs the geometric condition is slightly different. Since sticks are supposedly long and thin, their projections also have to be long and thin, ie the circularity of the region in the view has to be close to zero. Alternatively, we assume that blobs can only project onto regions of high circularity. This condition is enforced as a pair of thresholds that the measured circularity of the parts must satisfy.

Error of the mapping

Associated with the mapping from model parts to two-dimensional polygons is an error which specifies how well the model corresponds to the object in the view. The error is made up of two parts, structural error and completeness error as defined by Shapiro *et al*[23]. In our research, however, since the mapping is only one way (from the model to the polygons) the error formulae in that report[23] are reduced from two terms to just one. The effectiveness of this kind of error measure was demonstrated in the earlier report[23].

A tree search yields the mapping with the minimum structural error between the model and the decomposed view. Associated with the mapping is the computed estimate of the unknown camera position.

CONSISTENT LABELLING FORMALIZATION

The entire process of determining the lowest error

mapping between the primitives in the model and the decomposed parts of the image can be formalized as a consistent labelling problem as follows.

Let P be the set of primitives in the model. For each primitive $p \in P$, let $T(p) \in$ {stick, plate, blob} be the type of primitive p. Let CS be the connects/supports relation, and TR be the triples relation. Let S be the set of simple parts in the view, CS' be the two-dimensional connects relation, and TR' be the two-dimensional triples relation.

Let τ and ϕ be the sets of possible tilt and pan angles respectively. Let null be a special label to be used when a primitive in the model maps to no simple part in the view. Let C_1 be the circularity threshold for sticks and let C_2 be the circularity threshold for blobs.

An ε-consistent labelling is a mapping

$$f: P \rightarrow S \cup \{\text{null}\} \times \tau \times \phi$$

that satisfies the following three conditions.

Shape constraints

If $f(p) = (s, \tau\text{-p}, \phi\text{-p})$ and for some $x \neq p$, then

- if T(p) = stick, then $C(s) < C_1$
- if T(p) = blob, then $C(s) > C_2$
- if T(p) = plate, then $C(s) = \sin(\tau\text{-p})$.

This states that any stick in the model can only map onto a polygon in the image that has a circularity value less than a prespecified threshold, and blobs can only map to polygons with a high value of circularity. Note that the circularity measures are normalized to yield a value of 1.0 for circles and 0.0 for lines. Plates can map to features which have a circularity equal to the sine of the tilt angle predicted for the part.

View constraints

If

$$T(p_1) = \text{plate}$$
$$T(p_2) = \text{plate},$$
$$f(p_1) = (s_1, \tau\text{-}1, \phi\text{-}12)$$
$$f(p_2) = (s_2, \tau\text{-}2, \phi\text{-}21)$$

and $(p_2, p_1, \text{how21}, A, B, D) \in CS$, then τ-2 satisfies $E(\tau\text{-}1, \phi\text{-}12, A, B, D)$ and ϕ-21 satisfies $E(\tau\text{-}1, \phi\text{-}12, A, B, D)$ where E is the constraint propagation equation (described in the third section above) which relates the pan and the tilt angles on one plate to the pan and tilt angles of plates which it touches.

Relational constraints

Let

$$a = \sum_{i=1}^{\#P} \sum_{\substack{j=1 \\ j \neq i}}^{\#P} X_{ij}$$

where

$$X_{ij} = \begin{cases} 1 & \text{if } (p_i\ p_j \ldots) \in CS, \\ & f(p_i) = (s_i \ldots), \\ & f(p_j) = (s_j \ldots) \\ & \text{and } (s_i, s_j) \notin CS' \\ 0 & \text{otherwise} \end{cases}$$

Let

$$b = \sum_{\substack{k=1 \\ k \neq i}}^{\#P} \sum_{i=k}^{\#P} \sum_{\substack{j=1 \\ j \neq i}}^{\#P} Y_{kij}$$

where

$$Y_{kij} = \begin{cases} 1 & \text{if } (p_k\ p_i\ p_j \ldots) \in TR, \\ & f(p_k) = (s_k \ldots), \\ & f(p_i) = (s_i \ldots), \\ & f(p_j) = (s_j \ldots), \\ & (s_k, s_i, s_j \ldots) \notin TR' \\ & \text{or} \\ & \text{if } (p_k\ p_i\ p_j\ s_{ijk} \ldots) \in TR, \\ & f(p_k) = (s_k \ldots), \\ & f(p_i) = (s_i \ldots), \\ & f(p_j) = (s_j \ldots), \\ & C(s_i) < C_1, \\ & (s_k, s_i, s_k, s_{ijk}) \notin TR' \\ 0 & \text{otherwise} \end{cases}$$

Then

$$\frac{a + b}{\#CS + \#TR} \leq \varepsilon$$

This constraint set gives the error counting procedure for relational errors. It states that the total error is the sum of the number of relations of the model which fail to carry over to the image, normalized by the total number of relations in the model.

EXPERIMENTAL RESULTS

A database of 11 three-dimensional models was used as the source of the three-dimensional information for the mapping. These objects are shown in Figure 7. Two-dimensional views were either generated by a computer graphics system, from known camera positions, or obtained from digitized photographs. Nine views were generated for each object in the database at various pan angles around the object. The camera pan angle was changed in 20° increments from −90° to +70°. Because of the symmetry of most man-made objects, the views would repeat outside the range. Each of these views was decomposed using the two methods described in the previous section, and the resulting two-dimensional descriptions were compared with each model in the set. In each case the best mapping (that with minimum total error) was obtained along with an estimate of the camera position. The results are summarized below with two examples. The models shown in the examples are illustrated in Figure 8. Figure 8a shows a chair with arms. Figure 8b shows a four-legged table.

Graph-theoretic clustering results

For the two sample objects and their outlines shown in Figure 8, the decompositions generated by the clustering algorithm are shown in Figures 9a and 9b. The views clustered with the graph-theoretic clustering method matched the correct model in 67% of the cases. The failures were found in the cases when there was insufficient information in the outer boundary alone to enable a proper decomposition. One such case is shown in Figure 8a. This is a model of a chair whose arms are

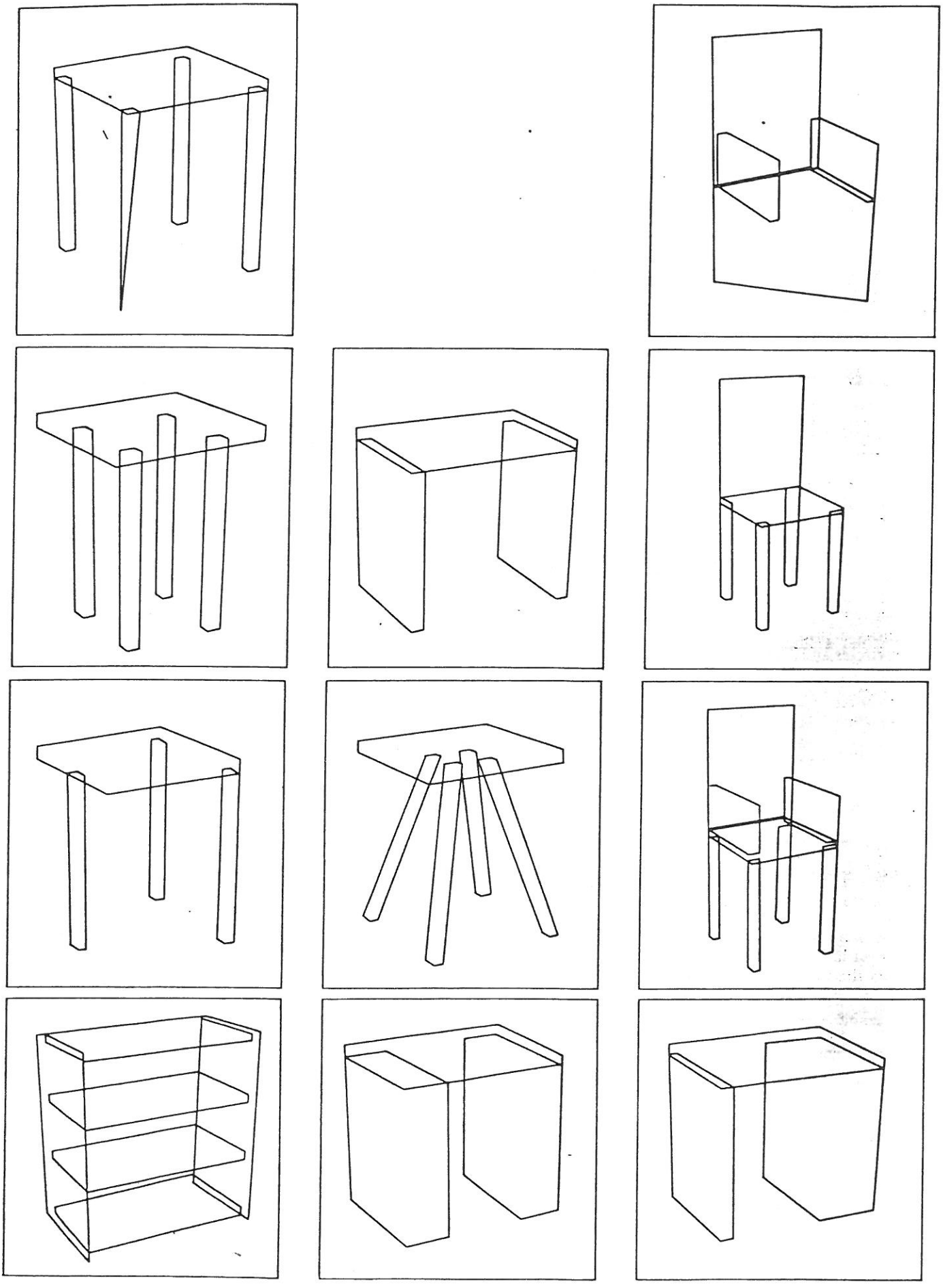

Figure 7. The 11 database objects used in testing the matching program

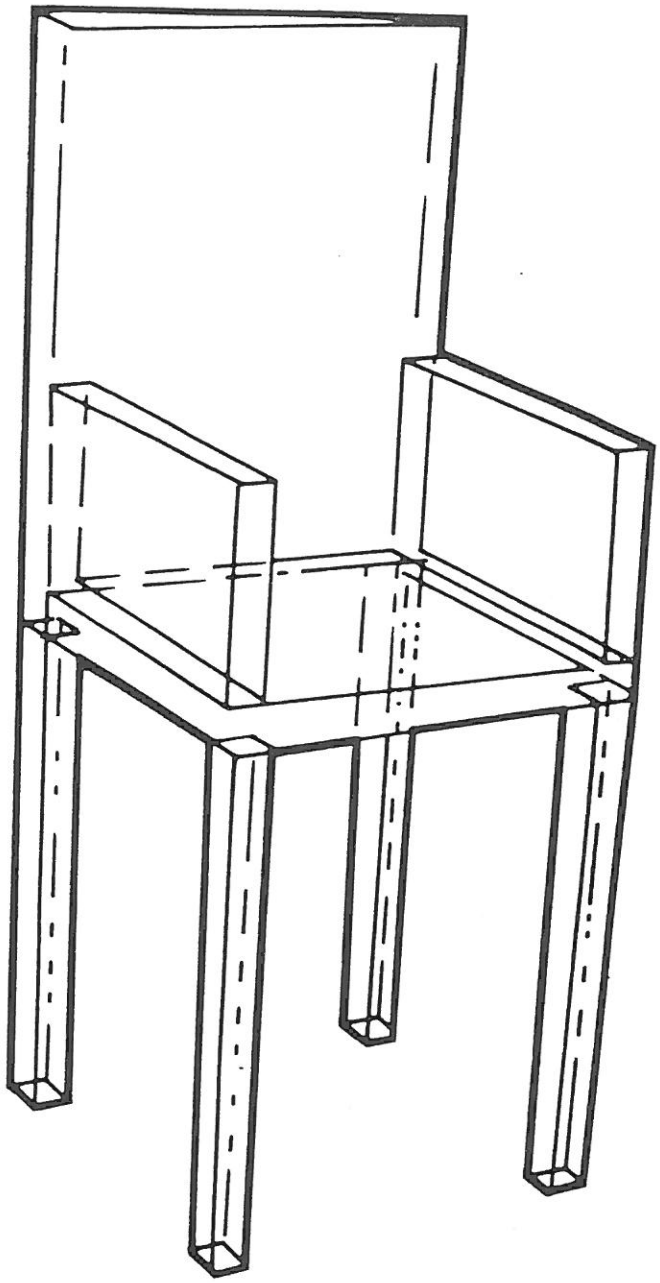

Figure 8a. Perspective view of a chair (the silhouette is shown in bold lines)

plate like. These arms obscure the structure of the chair behind it, and consequently the outer boundary fails to retain all necessary information. On the other hand, objects such as the table shown in Figure 8b provide enough information in their outer boundaries to enable accurate matches to be made.

Perfect decompositions

When views were decomposed on the basis of *a priori* knowledge of the boundaries of the parts, the success rate went up to 92%. The views that failed to match in these experiments were those containing blobs. Remember that blobs are idealized as spheres which we felt should show a high circularity in all orientations. The blobs in our models were more elongated, and in some orientations their circularity did not pass the threshold set for

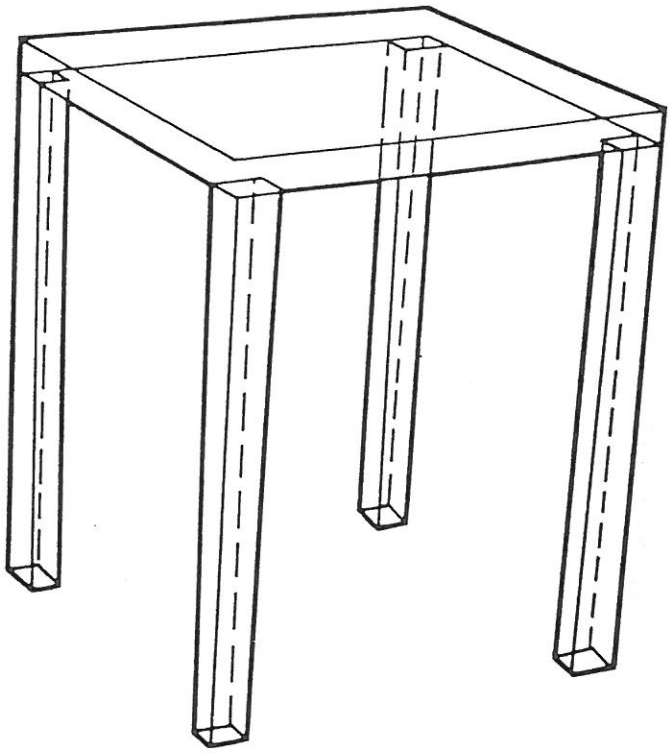

Figure 8b. Perspective view of a table (the silhouette is shown in bold lines)

blobs. The perfect decompositions of the objects shown in Figure 8 are illustrated in Figures 10a and 10b.

The angles computed for the camera positions were within 20° of the tilt and pan angles. Again the reason for the discrepancy is the difference between the idealized nature of the plate and the corresponding physical structure. The tilt angle is computed on the assumption that the part causing the associated projection is a circle. However, real plates (such as the tops of tables) are not always circular. Those in our database were rectangular. Projections of rectangles do not have unit circularity, even when the tilt angle is exactly 90.0°. Consequently the computed tilt angles are lower than the actual camera angles. This error then propagates into the pan angles. However, this is not a very large error, especially if more accurate matching methods are used to examine the models further.

On the average each view mapped to 3.4 and 2.0 models in each of the two sets of experiments. This is because the information on the sticks was not used for constraint satisfaction. Therefore objects which differed only in stick positions and in orientations would all map to the same view with the same error.

Figures 11a and 11b show the mapping between the two decomposed versions of the chair and table. A line is drawn from the three-dimensional primitive in the model to the two-dimensional part to which it mapped. In these figures the mapping error for each of the views is shown in parentheses as (structural error, completeness error). Parts which mapped to null are also indicated. Note that for the chair with arms, when the image was decomposed using the graph-theoretic clustering procedure, the model with the least error was a table with three legs because the arms did not show up in the outline and the back of the chair merged with the seat during clustering.

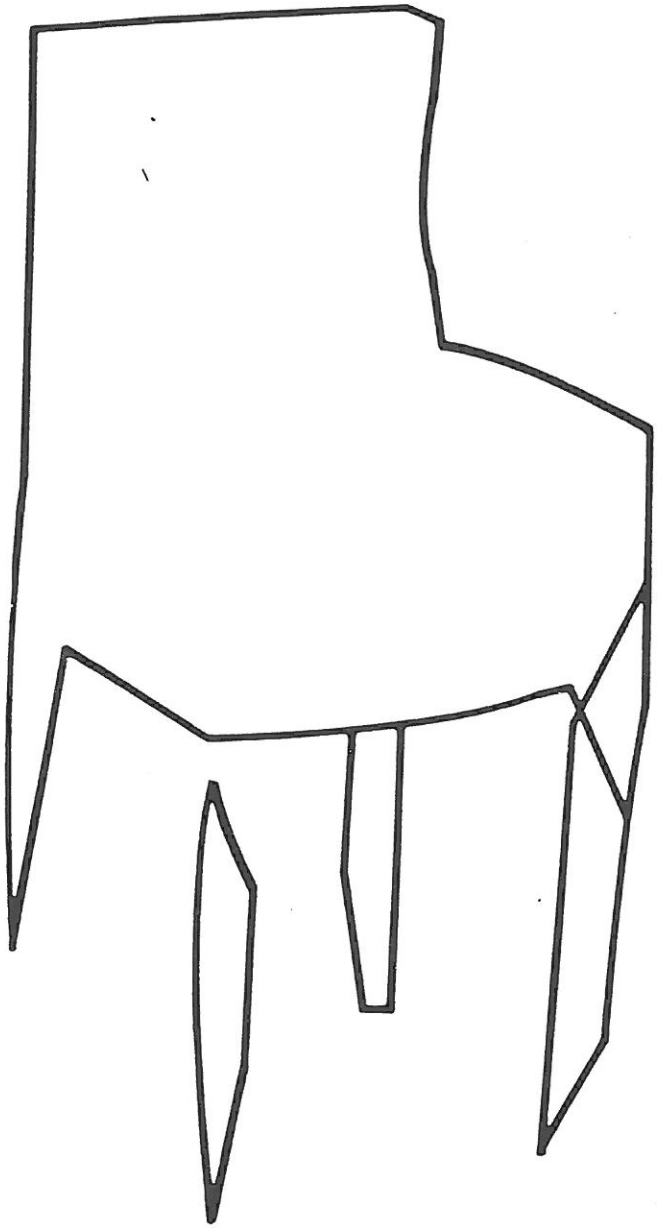

Figure 9a. Graph-theoretic decomposition of the silhouette in Figure 8a

CONCLUSIONS

In this paper we have demonstrated that, for some recognition purposes, rough three-dimensional models may contain sufficient information for matching. We have shown a technique by which rough models defining the structural and geometric relations in an object can be used in a scene analysis system. We have also shown how the geometric information can be used during the process of matching to constrain the possible interpretations for parts in the view, and how the camera location serves as a global constraint which reduces the possible interpretations for the scene.

We have shown experimentally that the mapping scheme is a robust method for analysing unknown views and that, with a proper front and capable of using more information for recognition decomposition, good results can be obtained.

It is clear that the outline of an object alone is not

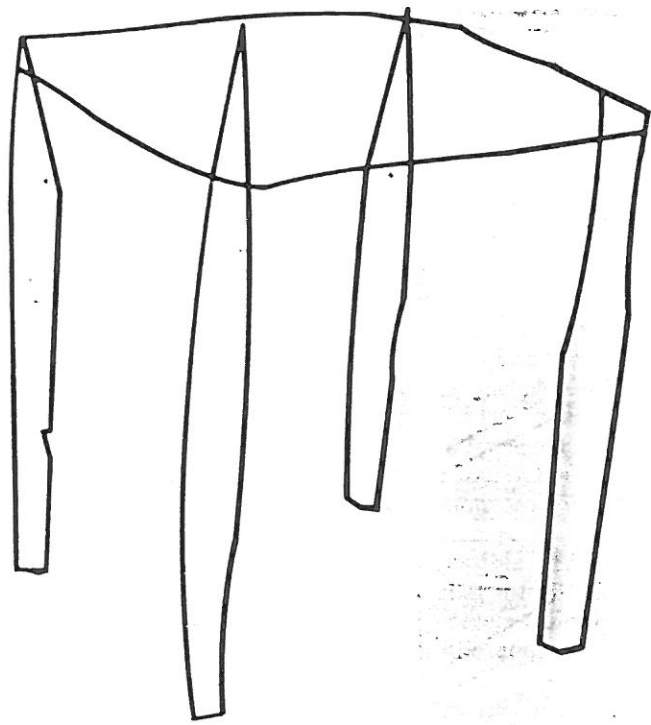

Figure 9b. Graph-theoretic decomposition of the silhouette in Figure 8b

enough to characterize it uniquely. Current research is aimed at using the information available in the foreshortening of sticks, and using all the information in greytone pictures for the extraction of the images of sticks, plates and blobs from the image.

ACKNOWLEDGEMENT

This research was supported by the US National Science Foundation.

REFERENCES

1 **Shapiro, L G, Mulgaonkar, P G, Moriarty, J D and Haralick, R M** 'A generalized blob model for three-dimensional object description' *2nd IEEE Workshop on Picture Description and Management* (August 1980)

2 **Voelcker, H B and Requicha, A A G** 'Geometric modelling of mechanical parts and processes' *Computer* Vol 10 No 12 (December 1977)

3 **Binford, T** 'Visual perception by computer' *IEEE Systems Science and Cybernetics Conf.* Miami, FL, USA (December 1971)

4 **Nevatia, R and Binford, T O** 'Description and recognition of curved objects' *Artif. Intell.* Vol 8 (1977)

5 **Marr, D and Nishihara, H K** 'Spatial disposition of edges in a generalized cylinder representation of objects that do not encompass the viewer' *Memo 341* MIT Artificial Intelligence Laboratory, MA, USA (December 1975)

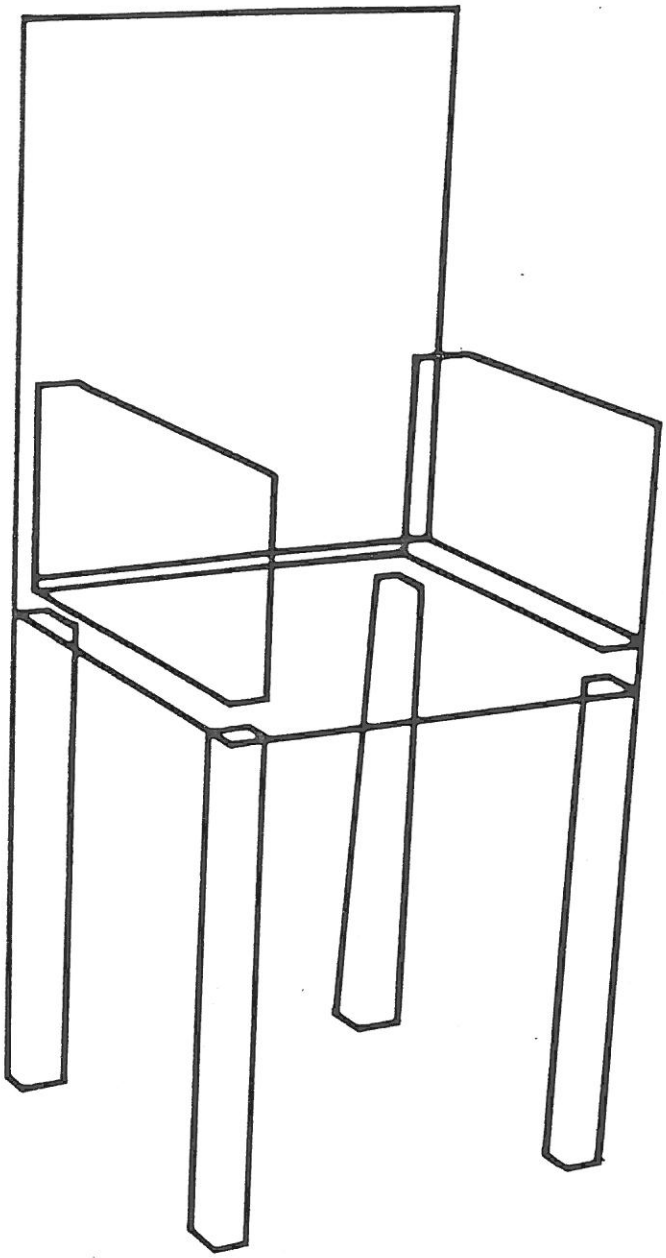

Figure 10a. Ideal decomposition of the object in Figure 8a

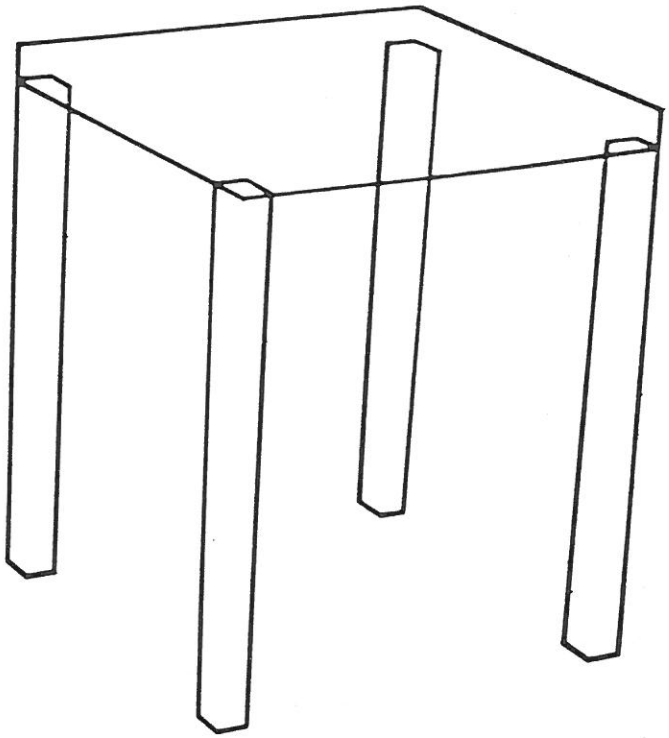

Figure 10b. Ideal decomposition of the object in Figure 8b

of two-dimensional shapes by graph-theoretic clustering' *IEEE Trans Pattern Anal. Mach. Intell.* Vol 1 No 1 (1979)

6 **Thomason, M G and Gonzalez, R C** 'Database representations in hierarchical scene analysis' in *Progress in pattern recognition* **Kanal, L N and Rosenfeld, A** (Eds) North-Holland, Amsterdam Netherlands (1981)

7 **Zhang Shouxuan** 'A chinese character recognition system based on pictorial database techniques' *Proc. 6th Int. Conf. on Pattern Recognition* Munich, FRG (1982)

8 **Shapiro, L G and Haralick, R M** 'Organization of relational models for scene analysis' *IEEE Trans. Pattern Anal. Mach. Intell.* Vol 4 No 6 (November 1982)

9 **Pavlidis, T** 'A review of algorithms for shape analysis' *Comput. Graphics Image Process.* (April 1978)

10 **Shapiro, L G and Haralick, R M** 'Decomposition

11 **Gennery, D B** 'A stereo vision system for an autonomous vehicle' *Proc. 5th Int. Joint Conf. on Artificial Intelligence* MIT, Cambridge, MA, USA (1977)

12 **Brooks, R A** 'Model-based three-dimensional interpretation of two-dimensional images' *Tech. Rep.* Stanford University, CA, USA (1980)

13 **Brooks, R A** 'Symbolic reasoning among three-dimensional models and two-dimensional images' *Artif. Intell.* Vol 17 (1981)

14 **Shapiro, L G** 'A structural model of shape' *IEEE Trans. Pattern Anal. Mach. Intell.* Vol 2 No 2 (March 1980)

15 **Moravec, H P** *Robot rover visual navigation* UMI Research Press (1981)

16 **Barrow, H G** 'Interactive aids for cartography and photo interpretation' *Semiannual Tech. Rep. of SRI Project 5300* SRI (October 1977)

17 **Fischler, M A and Bolles, R C** 'Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography' *Image Understanding Workshop* (May 1980)

18 **Rosenfeld, A, Hummel, R A and Zuker, S W** 'Scene labelling by relaxation operators' *IEEE Trans. Syst., Man Cybern.* Vol 6 No 6 (June 1976)

19 **Haralick, R M and Elliott, G** 'Increasing tree search efficiency for constraint satisfaction problems' *Proc. 6th Int. Joint Conf. on Artificial Intelligence* (1979)

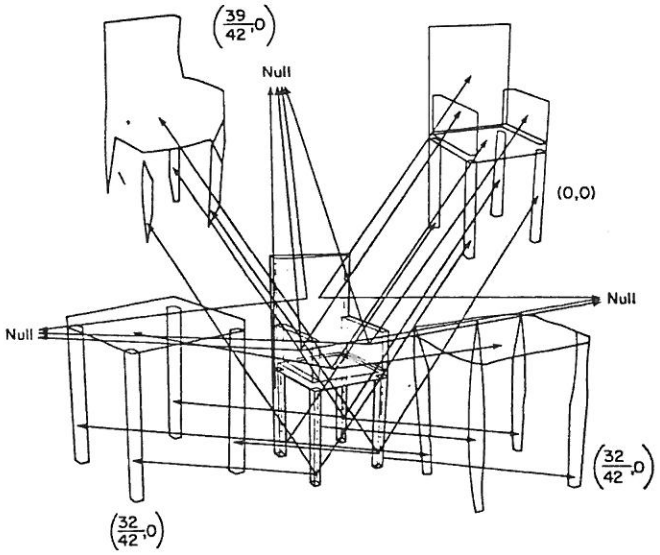

Figure 11a. The matching results for the chair object shown in Figure 8a: lines from the model part to the decomposed part show how the parts mapped; matching error is shown in parentheses as (structural error, completeness error)

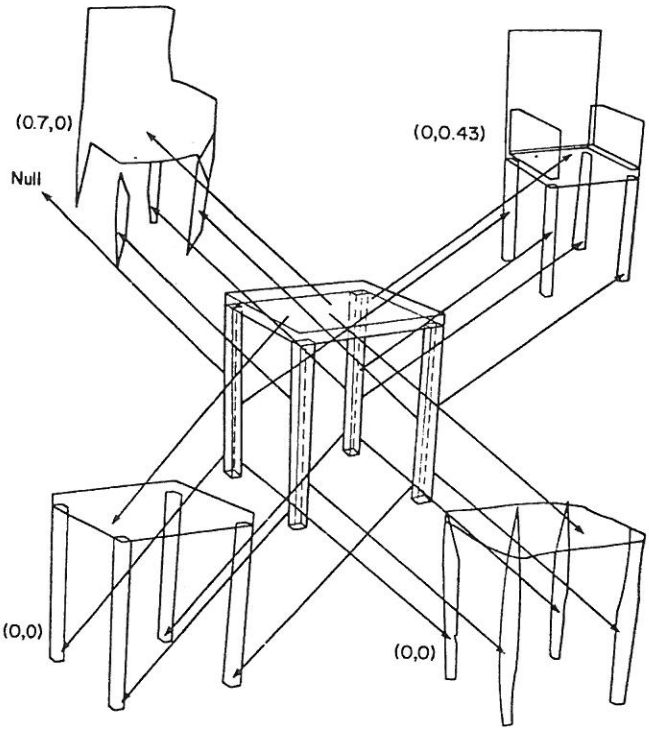

Figure 11b. The maching results for the table object shown in Figure 8b: lines from the model part to the decomposed part show how the parts mapped; matching error components are shown in parentheses as (structural, completeness)

20 **Mulgaonkar, P G** 'Recognition of three-dimensional objects from single perspective views' *Master's Thesis* Department of Computer Science, Virginia Polytechnic Institute and State University, VA, USA (December 1981)

21 **Haralick, R M, Laffey, T J and Watson, L T** 'The topographic primal sketch' *Int. J. Robotics Res.* Vol 2 No 1 (March 1983)

22 **Feustel, C D and Shapiro, L G** 'The nearest neighbor problem in an abstract metric space' *Pattern Recognition Lett.* Vol 1 No 2 (December 1982) pp 125–128

23 **Shapiro, L G, Moriarty, J D, Haralick, R M and Mulgaonkar, P G** 'Matching three-dimensional objects using a relational paradigm' *Tech. Rep. TR-CS80014R* Department of Computer Science, Virginia Polytechnic Institute and State University, VA, USA (December 1980; revised January 1983)

APPENDIX 1: ESTIMATION OF CAMERA PARAMETERS

Let us assume that we have two plates which touch edge to edge as described before. We shall show how the tilt and pan angles of the viewing vector measured with respect to one of the two plates are related to the tilt and pan angles measured with respect to the second plate. The camera location serves as a global constraint on the appearance of the various parts of the object. If we know what one of the parts of the object looks like, its appearance provides some information about the possible range of camea positions. Propagating this information to adjacent parts yields a system by which these constraints can be verified and used in narrowing down the possible range of camera locations.

Since our models do not have any global coordinate system, we cannot specify the camera position in an absolute sense. We can, however, specify the tilt and pan angles of the camera with respect to local part-centred coordinates for each part. We show in this section that, once we know the tilt and pan angles with respect to one plate, we can propagate them over to adjacent plates which it touches, and consequently over all connected groups of plates in the model.

Notation

We are given two plates U and V, which touch in an edge–edge type of connection (Figure 12), along with the three geometric constraints that form a part of the connects/supports relation. Let us also assume that we have selected one of the different possible physical configurations that could result from the given geometric values. The manner in which this configuration is decided will be described in later subsections.

The tilt angle (with respect to a specific plate) is the angle between the viewing vector **L** and the plane of the plate. The pan angle is the angle between the projection of the viewing vector onto the plane of the plate and the vector from the centre of the plate to the point of contact with some other prespecified three-dimensional part. This means that the pan angle is specified not merely with respect to a given plate but also with respect to its contact with some other specified plate.

Let the connection of U and V be reported as (V, U, edge–edge, *A, B, D*) where *A, B* and *D* are the three angles required for the specification of the connection. *A* is the angle between the vector from the point of contact between the plates to the centre of the plate V and its projection onto the plane of plate U. *B* is the angle

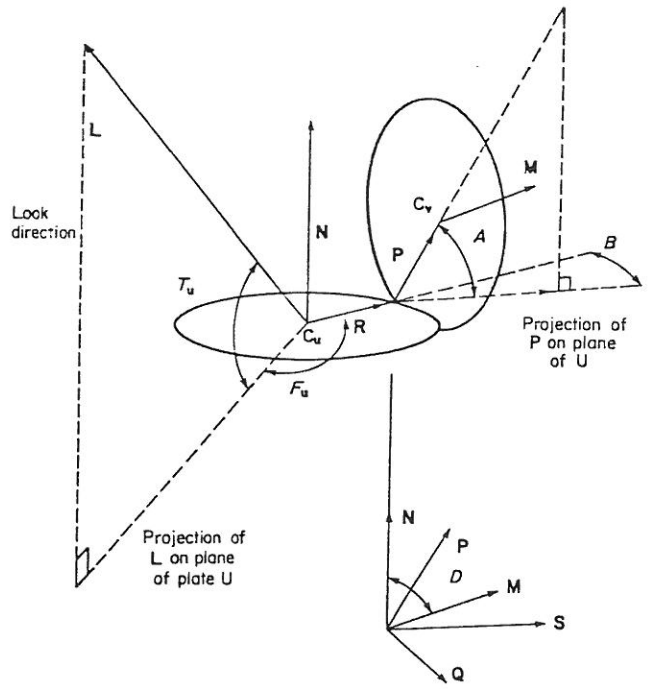

*Figure 12. Edge–edge connection of two plates U and V showing all relevant vectors and angles: C_v, centre of plate V; C_u, centre of plate U; **N**, normal to plate U; **M**, normal to plate V; **R**, vector from C_u to the point of contact; **P**, vector from the point of contact to C_v*

between this projection and the vector from the centre of plate U to the point of contact. D is the angle between the normals of the two plates. All these angles are indicated in Figure 12.

Let C_v be the centre of plate V, C_u be the centre of plate U, **N** be the normal to plate U, and **V** be the normal to plate V. Similarly let **R** be the vector from C_u to the point of contact and **P** be the vector from the point of contact to C_v. In the equations which follow, all capital letters refer to vectors or angles, and subscripts x, y and z refer to the projections of the vectors on the X, Y and Z directions respectively. For example, P_x is the X component of the vector **P**. Let F and T be the pan and the tilt angles respectively. These are qualified by the letters u and v to denote the plate with respect to which they are being measured.

All vectors are assumed to be unit vectors. $\times$ refers to vector cross products and '.' refers to the vector inner product. Multiplication between scalars is implicit, ie P_xN_y represents the scalar multiplication of P_x and N_y. All angles are expressed in degrees.

Computations

We wish to show that, given the tilt and pan angles T_u and F_u with respect to plate U and the connection angles A, B and D, we can compute the tilt and pan angles T_v and F_v with respect to plate V. To do that, we show how all the vector directions can be expressed in terms of the given angles A, B, D, T_u and F_u. Once we know the directions for all vectors in Figure 12, calculation of the required tilt and pan angles is straightforward.

We define three auxilary angles A', T'_u and T'_v to be $90.0° - A$, $90.0° - T_u$ and $90.0° - T_v$ respectively. We first determine a unit vector **S** which lies along the direction of the projection of the vector **P** onto the plane of plate U

$$\mathbf{Q} = \frac{\mathbf{P} \times \mathbf{N}}{\sin A'}$$

$$\mathbf{S} = \mathbf{N} \times \mathbf{Q}$$

The division by $\sin A'$ in the definition of **Q** makes **Q** a unit vector. Since **N** and **Q** are both unit vectors, and they are orthogonal, **S** is also a unit vector. Since **Q** is the cross product of the vectors **P** and **N**, it is perpendicular to the plane containing the two vectors. Moreover, since **N** is normal to the plane of the plate U, **Q** lies in the plane of U. Now **S** is normal to both **N** and **Q**. Consequently it must lie in the intersection of the plane containing **P** and **N** and the plane of the plate U. Therefore it is the projection direction of vector **P** in the plane of the plate U.

Using the conventions defined earlier, we have

$$\cos A' = \mathbf{N}.\mathbf{P}$$
$$\cos B = \mathbf{S}.\mathbf{R}$$
$$\cos D = \mathbf{N}.\mathbf{M}$$

The equations for the vectors **Q** and **S** may be expanded in terms of their components in the prime directions to obtain expressions for Q_x, Q_y, Q_z and S_x, S_y, S_z. These expressions can be substituted in the expression for $\cos B$ to yield

$$\cos B = \mathbf{S}.\mathbf{R} = (\mathbf{N} \times \mathbf{Q}).\mathbf{R}$$

which when expanded yields

$$\begin{aligned}\cos B = \{&[N_y(P_xN_y - P_yN_x) - N_z(P_zN_x - P_xN_z)]R_x \\ &+ [N_z(P_yN_z - P_zN_y) - N_x(P_xN_y - P_yN_x)]R_y \\ &+ [N_x(P_zN_x - P_xN_z) - N_y(P_yN_z - P_zN_y)]R_z \\ &\times (\sin A')^{-1}\end{aligned}$$

The angles T'_u and T'_v are defined by the expressions

$$\cos T'_u = \mathbf{L}.\mathbf{N}$$
$$\cos T'_v = \mathbf{L}.\mathbf{M}$$

The angle F_u is defined as the angle between the projection of the viewing vector onto the plane of plate U and the vector **R**. To obtain the vector **F** which is the projection of **L** on U, we proceed in the same fashion as we did for the projection of **P** on U

$$\mathbf{E} = \frac{\mathbf{N} \times \mathbf{L}}{\sin T'_u}$$

$$\mathbf{F} = \mathbf{E} \times \mathbf{N}$$

F is now the projection of the unit vector **L** on the plane of U. We can now generate the expression for F_u as the inverse cosine of the dot product of the vectors **F** and **R**, ie

$$\begin{aligned}\cos F_u &= \mathbf{F}.\mathbf{R} \\ &= (\mathbf{E} \times \mathbf{N}).\mathbf{R} \\ &= \frac{[(\mathbf{N} \times \mathbf{L}) \times \mathbf{N}].\mathbf{R}}{\sin T'_u}\end{aligned}$$

Similarly, by considering the projection of the vector **L** onto the plane of V, we get

$$\cos F'_v = \frac{[(\mathbf{M} \times \mathbf{L}) \times \mathbf{M}] . (-\mathbf{P})}{\sin T'_v}$$

The terms involving the double cross products can be expanded to obtain the expression for the angles in terms of the components of the vectors. The expression for cos F_v, for example, becomes

$$\begin{aligned}\cos F_v = {} & \{[M_z L_x - M_x L_z)M_z - (M_x L_y - M_y L_x)M_x]P_x \\ & + [(M_x L_y - M_y L_x)M_x \\ & \quad - (M_y L_z - M_z L_y)M_y]P_y \\ & + [(M_y L_z - M_z L_y)M_y \\ & \quad - (M_z L_x - M_x L_z)M_z]P_z\} \\ & \times (\sin T'_u)^{-1}\end{aligned}$$

The calculations so far were independent of the choice for the coordinate directions. Since the choice of the coordinate system is arbitrary, we can select the system which simplifies all expressions involved. Specifically, let us choose a right-handed coordinate system such that the vector **N** becomes (0, 0, 1) and the vector **R** becomes (0, 1, 0). In this coordinate system the expressions for the angles become

$$\begin{aligned}\cos B &= P_y / \sin A' \\ \cos A' &= P_z \\ \cos D &= M_z \\ \cos T'_u &= L_z \\ \cos F_u &= L_y / \sin T'_u\end{aligned}$$

Out of the five vectors **M, N, R, P** and **L**, the vectors **M, P** and **L** were unknown. **N** and **R** were defined by our choice of coordinate system above. However, the lengths of these three vectors are known (to be unity). So another constraint is that the sum of the squares of each component equals unity for each vector. Moreover, since **M** and **P** are orthogonal, we get the equation

$$M_x P_x + M_y P_y + M_z P_z = 0$$

Therefore we can explicitly solve for the values of the components of **M**. This means that the entire connection geometry is defined. We can determine the vectors **M, N, P** and **R** in terms of the angles *A, B* and *D*.

The vector **L** is also a unit vector, and its components are involved in the expressions for the camera constraints — the tilt and pan angles

$$\begin{aligned}L_x &= \sin T'_u \sin F_u \\ L_y &= \cos F_u \sin T'_u \\ L_z &= \cos T'_u\end{aligned}$$

The explicit solution of the vectors reveals that, since the signs of the angles *A, B* and *D* are undefined, we have up to eight different solutions. For the purpose of this section we assume that, from these eight, one solution has been extracted. In practice this choice is made by propagating the constraints on the camera tilt and pan angles for each of the eight solutions; the wrong solutions often give rise to inconsistencies, leaving only the correct solution. For certain values of the angles, several of these solutions collapse into a single value (multiplicity of roots of the defining equations), reducing the search space.

Because of symmetries in the structures of some of the objects, their appearance from several distinct viewpoints may be the same. For example, the pan angle makes no difference to the projection of an isolated plate. Similarly the views of a symmetric four-legged table remain unchanged if the pan angle is incremented in multiples of 90°. The sets of camera positions from which the view of the object appears the same form equivalence classes partitioning the space of possible viewing locations. In the absence of any external information about camera position, the words 'correct solution' should be interpreted as 'member of the equivalence class to which the correct solution belongs'.

Once all the vectors shown in Figure 12 have been defined, computation of the required angles T'_v and F'_v is trivial. Their values have already been defined in terms of vector dot products earlier. Moreover, once these vectors have been fixed we can compute the angles that need to be specified for the U, V connection. Remember that the angles in a U, V connection are not necessarily the same as the angles in a V, U connection. However, these angles are once again definable in terms of the dot products of vectors already computed.

BIBLIOGRAPHY

Davis, L S 'Shape matching using relaxation techniques' *IEEE Trans. Pattern Anal. Mach. Intell.* Vol 2 No 3 (March 1981)

Laffey, T J, Haralick, R M, Mulgaonkar, P G and Shapiro, L G 'A one pass border tracking algorithm' *Tech. Rep. CS81018-R* Department of Computer Science, Virginia Polytechnic Institute and State University, VA, USA (1981)

Lanfue, G 'Recognition of three-dimensional objects from orthographic views' in **Pooch, U W** (ed.) *Proc. 3rd Annu. Conf. on Computer Graphics and Image Techniques and Information Processing* (1976)