Using Rough Relational Models for Geometric Reasoning

by

Prasanna G. Mulgaonkar, Linda G. Shapiro, Robert M. Haralick

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

ABSTRACT

The use of coarse volumetric, geometric and relational models for the analysis of perspective projections of solid objects is discussed. Using these models, equations are given for the computation and estimation of the camera parameters under which the perspective projection image was created. A matching scheme is developed for the process of identifying an unknown object in a scene.

Keywords: Scene Analysis, Three-Dimensional, Two-Dimensional, Modelling, Recognition, Geometry, Relations.

I. Introduction

A computer system is given a single perspective view of an unknown object, taken from an unspecified camera position. Its goal is to describe and possibly identify the object in the scene. Our experiments deal with a class of man-made objects including furniture such as chairs, tables, desks, etc. The methods presented in this paper could be extended to include other types of objects in a relatively straightforward manner.

Human beings have the capacity to perform visual recognition tasks with relative ease and often show very good performance even when the picture is degraded by noise. Depending on the camera geometry and the lighting of the scene, parts of the object may be occluded or shaded to the extent that object identification becomes extremely difficult. The ability to extract the significant parts of the image, and to reason about the geometric and spatial relations between these extracted parts seems to play a definite role in the process of recognition. In an effort to explain, and possibly achieve the level of human efficiency, various researchers have favoured the idea that analysis of scenes relies on the presence of a-priori knowledge about the scene and the nature of the objects likely to be found in it.

This research was supported by the National Science Foundation under grants MCS-7923827, MCS 7919741 and MCS-8102872.

This paper discusses a modeling scheme for describing three-dimensional objects, and the geometric reasoning that can be done using the information stored in these models.

II. Literature Review

We first present a brief review of some of the other work being done in this area by other researchers. This is by no means a complete survey. It is mainly intended to be indicative of the variety of techniques being examined.

Object Modeling

There exists a large volume of work which has been reported in the field of three-dimensional object modeling and representation. Most of the current techniques build up descriptions of objects from simpler primitives in various ways. Constructive solid geometry (VOE77) systems use set-theoretic "additions" and "subtractions" of solid primitives to assemble objects. Binford (BIN71) first proposed a scheme of decomposing objects into "generalized cylinders". The generalized cylinder modeling was incorporated in a system for performing scene analysis experiments by Nevatia (NEV77). This technique was extended to a hierarchic system by Marr and Nishihara (MAR75). The modeling scheme used in our work is the "generalized blob" model proposed by Shapiro et.al. (SHA80a).

Image Decomposition

Decomposition of images into "meaningful" parts has also received a considerable amount of attention. Conceptually the parts that are obtained, should correspond to projections of the three-dimensional primitives used for modeling. Two-dimensional shape decomposition algorithms have been reported by several researchers. 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 (SHA79).

Matching Schemes

Relational matching of polygonal shapes has been reported by Shapiro (SHA80b). Brookes (BRO80,BRO81) used symbolic reasoning in the context of recognizing three-dimensional objects from single perspective views. Relational matching traditionally has required large exponential-time searches. The use of discrete relaxation for the matching process was formalized by Rosenfeld et.al. (ROS76). Later Haralick and Elliott (HAR79) examined speedups and tree pruning techniques that could be used for speeding up the tree searches used in such matching processes.

III. Generalized Blob Models

The modeling scheme used in our work, is the generalized blob scheme developed by Shapiro et.al. (SHA80a). This technique has been described in detail in (MUL81). In this paper we will only 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 modeling 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. Sticks are inherently linear features, like chair legs, and are modeled as straight lines in three-space. Plates are the flat parts like chair backs or table tops. These are modeled as circular disks in three space. Blobs are modeled as spheres and are the parts that occupy a large volume.

The model contains a list of the parts of the object, along with the geometric and relational interactions among them. There are three main relations which have been used for the task of object recognition: unary, binary, and the ternary relations. The unary relation describes the relative lengths, areas, volumes, and types of each of the primitives. Of course, sticks do not have any area or volume, just as plates do not have any volume. All these characteristics are relative to the smallest part in that particular model. This makes the description of the objects invariant under global scale changes.

The binary and the ternary relations are the most important relations in a model. These relations describe how the primitive parts fit together to make up the structure of the object. The binary CONNECTS/SUPPORTS relation describes pairs of touching primitives by specifying the spatial relationships in terms of symbolic points and the geometric relationships in terms of the angles formed between the various significant axes of the primitives involved. All measurements are specified with respect to local part-centered coordinate systems. This makes the object description translation and rotation independant. We describe the encoding of the CONNECTS/SUPPORTS relation in detail in Section 4.3.

The TRIPLES relation makes explicit the nature of subassemblies of three parts. For every set of three primitives, A, B and C, such that both parts A and C touch the middle part B, the TRIPLES relation specifies the spatial relationship of A and C to B, along with one angle necessary to fix the geometry. This angle is the one formed between the two lines joining the centers of parts A and C, to the center of B. A simple example involving three sticks is shown in Figure 1. This information 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.

The Geometry of a Binary Connection

In this section we will examine the way in which a binary connection would be encoded in the database. In particular, we will study the nature of a plate-plate connection in which the edges of the plates touch each other. This kind of a contact occurs very often in man-made objects. For example, the back of a chair and the seat touch in that way (Figure 8).

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

Since the connection is described in terms of a local coordinate frame, the coordinate directions can be chosen so as to make the analysis as simple as possible. In particular, we choose the origin at the center 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 center of A to the point of its contact with B.

Since the radius of B is known, the center of B can lie anywhere on a sphere centered at P (the point of contact between A and B). Think of a polar coordinate system centered at P. The center of B has two degrees of freedom, and therefore two angles are enough to describe the ray from P to the center of B. The plane of the plate B has one more degree of freedom - all it has to do is pass through the ray just fixed. Consequently three angles are enough to characterise the entire plate-plate edge-edge geometry.

The three angles actually measured, are shown in Figure 2. Angle alpha is the elevation of the center of the second plate from the first, beta is the swing angle of the second plate, and delta 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 parts B and A. The reason for this is that the angles are specified in terms of coordinate systems centered 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. Alpha can lie between 0 and 90 degrees, beta from 0 to 180 degrees, and delta between -90 and +90 degrees. Note that alpha and beta do not have signs. Since there is no global coordinate system, there is no way of specifying clockwise or anti-clockwise rotations. This is precisely what makes the description insensitive to mirror image reflections. However because of this feature, if the three angles are given, up to eight physical interpretations can be constructed. Figure 3 shows the four different orientations of the ray from P to the center of B given the angles alpha and beta. For each of these orientations, there are two possible orientations for the angle delta. Figure 3 also shows how these eight descriptions are related - they are reflections about the XZ plane. i.e. the plane containing the normal to A and the line from the center of A to P.

At the expense of 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 (MUL81).

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 the actual 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 in screen coordinates of the point, can be expressed as

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

The camera itself is located with its origin at Xc, Yc and Zc in the world coordinate system. Without loss of generality, 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 translation 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 projective transformation. Every point in the scene is the image 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 information than what is 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 projection involved. In particular, we can assume that the swing angle is zero and 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 the focal length of the lens is large. The resulting projection is called a perspective normal projection (c.f. Brookes (BRO81)), because it is the equivalent of a normal projection onto a plane which is parallel to the screen and close to the object, folowed 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 under perspective normal projection ? Blobs have been described as spheres in three space. The normal projection of a sphere on any arbitrary plane, is simply a circle. Sticks project as lines (or depending on the viewpoint, vanish). Plates are the most interesting since their projection yields the most information about the relationship of the camera to the object.

Plates are modeled 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 theta (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 center of our object, the scale factor is close to 1.0. Note that the angle theta in Figure 6 is 90.0 - the tilt angle shown in

Figure 5. That means that the projection of a plate yields some information about the picture taking process. How this information can be utilized is the subject of subsequent sections.

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

In real objects, plates are not always exact circles. For example, the tops of tables (which would be modeled 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 will be discussed in later sections.

Estimation of Camera Parameters

Let us assume 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 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 some absolute sense. We can however, specify the tilt and pan angles of the camera with respect to local part centered coordinates for each part. In fact, we will 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 to all parts in the model.

Some Notation We are given two plates U and V, which touch in an edge-edge type connection (Figure 7), 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 way in which this configuration is decided, will be described in later sections.

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 view vector on the plane of the plate, and the vector from the center of the plate to the point of contact with some other pre-specified 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 center of plate V and its projection on the plate U. B is the angle between this projection and the vector from the center of the 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 7.

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

All vectors are assumed to be unit vectors. "X" refers to vector cross products and "." refers to the vector inner products. Multiplication between scalars is implicit. i.e. PxNy represents the scalar multiplication of Px and Ny. All angles are implicitly in degrees.

The Computations We wish to show that given the tilt and pan angles Tu and Fu with respect to the plate U, and the connection angles A, B and D, we can compute the tilt and pan angles Tv and Fv 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, Tu and Fu. Once we know the directions for all vectors in Figure 7, calculation of the required tilt and pan angles is simple.

We first define three auxilliary angles A', T'u and T'v to be 90.0 - A, 90.0 - Tu and 90.0 - Tv respectively. We first wish to determine a unit vector S which lies along the direction of the projection of the vector P on the plane of U.

$$Q = P \times N / \sin(A')$$
$$S = N \times Q$$

The division by sin(A') in the definition of Q makes Q a unit vector. Since N and Q are now both unit vectors and orthogonal, S is also a unit vector. Since Q is the cross project of the vectors P and N, it is perpendicular to the plane containing the two vectors. Also 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:

$$\cos(A') = N \cdot P$$
$$\cos(B) = S \cdot R$$
$$\cos(D) = N \cdot M$$

The equations for the vectors Q and S may be expanded in terms of their components in the prime directions to obtain expressions for Qx, Qy, Qz and Sx, Sy and Sz. These expressions can be substituted in the expression for cos(B) and the expression expanded to yield:

$$\cos(b) = S.R = (NxQ).R$$

$$\cos(B) = \frac{\begin{array}{l}(Ny(PxNy-PyNx) - Nz(PzNx-PxNz))Rx \\ +(Nz(PyNz-PzNy) - Nx(PxNy-PyNx))Ry \\ +(Nx(PzNx-PxNz) - Ny(PyNz-PzNy))Rz\end{array}}{\sin(A')}$$

The angles T'u and T'v are defined by the expressions:

$$\cos(T'u) = L \cdot N$$
$$\cos(T'v) = L \cdot M$$

The angle Fu is defined as the angle between the projection of the viewing vector on the plane of the 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.

$$E = N \times L / \sin(T'u)$$
$$F = E \times N$$

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

$$\begin{aligned}\cos(Fu) &= F.R \\ &= (E \times N).R \\ &= ((N \times L) \times N).R / \sin(T'u)\end{aligned}$$

and similarly by considering the projection of the vector L on the plane of V, we get:

$$\cos(F'v) = ((M \times L) \times M).-P/\sin(T'v)$$

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(Fv), for example, becomes:

$$\cos(Fv) = \frac{\begin{array}{l}((MzLx-MxLz)Mz - (MxLy-MyLx)Mx)\ Px \\ +((MxLy-MyLx)Mx - (MyLz-MzLy)My)\ Py \\ +((MyLz-MzLy)My - (MzLx-MxLz)Mz)\ Pz\end{array}}{\sin(T'u)}$$

These calculations so far were independant of the choice for the coordinate directions. Since the choice of the coordinate system is arbitrary, we can select the one 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:

$$\cos(B) = Py/\sin(A')$$
$$\cos(A') = Pz$$
$$\cos(D) = Mz$$
$$\cos(T'u) = Lz$$
$$\cos(Fu) = Ly/\sin(T'u)$$

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 is known (to be unity). So another constraint is that the sum of the squares of each component, totals to one for each vector. Also since M and P are orthogonal, we get the following equation:

$$MxPx + MyPy + MzPz = 0$$

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.

$$Lx = \sin(T'u)\sin(Fu)$$
$$Ly = \cos(Fu)\sin(T'u)$$
$$Lz = \cos(T'u)$$

The explicit solution of the vectors reveals the fact that since the signs of the angles A, B and D are undefined, we have upto 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 and the wrong solutions give rise to inconsistancies, 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.

Once all the vectors shown in Figure 7 are defined, computation of the required angles T'v and F'v are trivial. Their values have already been defined in terms of vector dot products earlier. There is an added bonus that results in this computation. Once the vectors are 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. But these angles are once again definable in terms of the dot products of vectors already computed.

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 acts 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 constraints have to meet the consistency checks described below.

Geometric Constraints As described before, the geometric constraint arises from the fact that a single camera position generates the entire image. This is imposed by requiring that the when the propagation of constraint angles yields a value for the estimated tilt angle of any plate, it has to 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. Of course, this is not a two way implication. Parts that touch in the view do not neccesarily correspond to connected three-dimensional primitives.

Putting It All Together The entire matching strategy can now be formalized. The process consists of finding a consistent interpretation for all the visible parts in the view. This necessitates a preprocessing phase in which the image is decomposed into constituent two-dimensional regions (possibly overlapping) which intuitively, should correspond to the projections of the models three-dimensional parts.

Two-Dimensional Pre-Processing We worked with a graph-theoretic clustering procedure (SHA79) developed for generating near-convex clusters. The input is an ordered sampling of the points on the outer and inner (hole) boundaries of the sillhouette of the projection of the object. Of course, this method does not retain any of the detail of lines interior to the sillhouette and consequently, it does not work well in all possible cases. However, for many models, the outer boundary retains 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 are described later.

The Matching Process Once a decomposition is obtained, comparison with a model proceeds as follows. For every three-dimensional part, we try to select a decomposed 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 instantiation by comparing the predicted tilt angle with the computed value. If the values do not lie in the predicted range, that mapping is ruled out. Of course, 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, i.e. the circularity of the region in the view has to be close to 0. Alternately 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 is an error which specifies how well the model corresponds to the object in the view. The error is made up of two parts. The first part called the structural error, quantifies the difference between the relational structures of the model and the object in the view. It is the normalized sum of the total number of binary and ternary relations that fail to map over to the two-dimensional view. The second part of the error is the completeness error which provides a numerical measure of the completeness of the mapping. It is the fraction of the total number of parts in the view that were not the projected image of any model part. The completeness error also indicates any mismatch between the sizes of the model and the unknown object in the view. For example, since the structure of a table is a subset of the structure of a chair, the structural error between such a pair would be zero. However, the completeness error will indicate that the chair has one part more than the table (the back).

The total error of any mapping is the arithmetic mean of these two errors and shows how close the model is to the object in the image. The tree search yields the minimum structural error mapping between the model and the decomposed view. Associated with the mapping is the computed estimate of the unknown camera position.

IV. Experimental Results

The techniques outlined in the previous chapters, were implemented in RATFOR (a structured dialect of FORTRAN) and were tested on a variety of data. A database of eleven three-dimensional models was used as the source of the three-dimensional information for the mapping. Two dimensional views were generated either by computer graphics system, from known camera positions, or obtained from digitized photographs. Nine views were generated for each object in the database at varying pan angles around the object. The camera pan angle was changed in 20 degree increments from

-90 degrees to +70. Because of the symmetry of most manmade objects, the views would repeat outside this range. Each of these views were decomposed using the two methods described in the previous section, and the resulting "clusters" were compared with each model in the set. In each case the best mapping (one with minimum total error) was obtained along with an estimate of the camera position. The results are summarized below.

Graph-Theoretic Clustering Results

The views clustered with the graph-theoretic clustering method matched the correct model in 67 percent 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 8. This is a model of a chair whose arms are 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 8, provide enough information in their outer boundaries to enable accurate matches to be made.

Perfect Decompositions When views were decomposed based on a-priori knowledge of the boundaries of the parts, the success rate went up to 92 percent. The views that failed to match in these experiments were those containing blobs. Remember that blobs are idealized as spheres which 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 blobs.

The angles computed for the camera positions were within 20 degrees of the tilt and pan angles. The reason for the discrepancy, again, 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. The ones in our database were rectangular. Projections of rectangles do not have unit circularity, even when the tilt angle is exactly 90.0 degrees. 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 further examine the models.

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 orientations, would all map to the same view with the same error.

V. Conclusions

In this paper, we have demonstrated a technique by which rough three-dimensional 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 experimentally shown that the mapping scheme is a robust method for analysing unknown views and that, with a proper front end capable of using more information for region decomposition, good results can be obtained.

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.

REFERENCES

1. [BIN71] Binford, T., "Visual Perception by Computer", IEEE Systems Science and Cybernetics Conference, Miami, Florida, December 1971.

2. [BRO80] Brookes, R.A., "Model Based Three-Dimensional Interpretation of Two-Dimensional Images", Technical Report, Stanford University, 1980.

3. [BRO81] Brookes, R.A., "Symbolic Reasoning Among Three-Dimensional Models and Two-Dimensional Images", Artificial Intelligence, Special Volume on Computer Vision, AI 17, 1981.

4. [DAV79] Davis, L.S., "Shape Matching Using Relaxation Techniques", IEEE Transactions on Pattern Analysis and Machine Intelligence", PAMI-2, No. 3, Mar 81.

5. [HAR79] Haralick, R.M., Elliott, G., "Increasing Tree Search Efficiency for Constraint Satisfaction Problems", Proceedings of th Sixth International Joint Conference on Artificial Intelligence, 1979.

6. [LAF81] Laffey, T.J, Haralick, R.M., Mulgaonkar, P.G., Shapiro, L.G., "A One Pass Border Tracking Algorithm", Technical Report CS81013-R, Department of Computer Science, Virginia Polytechnic Institute and State University, 1981.

7. [LAN76] Lanfue, G., "Recognition of Three-Dimensional Objects from Orthographic Views", Proceedings of the Third Annual Conference on Computer Graphics and Image Techniques and Information Processing, Pooch, U.W. (Ed.), 1976.

8. [MAR75] Marr, D., Nishihara, H.K., "Spatial Disposition of Edges in a Generalized Cylinder Representation of Objects that do not Encompass the Viewer", MIT AI Lab Memo #341, Dec 1975.

9. [MUL81] 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, Dec 1981.

10. [NEV77] Nevatia, R., Binford T.O., "Description and Recognition of Curved Objects", Artificial Intelligence -8, 1977.

11. [ROS76] Rosenfeld, A., Hummel, R.A., Zuker, S.W., "Scene Labeling by Relaxation Operators", IEEE Transactions on Systems, Man and Cybernetics, CMS-6, No. 6, June 1976.

12. [SHA79] Shapiro, L.G., Haralick, R.M., "Decomposition of Two-Dimensional Shapes by Graph-Theoretic Clustering", IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1, No. 1, 1979.

13. [SHA80a] Shapiro, L.G., Mulgaonkar, P.G., Moriarty, J.D., Haralick, R.M., "A Generalized Blob Model for Three-Dimensional Object Description", Second IEEE Workshop on Picture Description and Management, August 1980.

14. [SHA80b] Shapiro, L.G., "A Structural Model of Shape", IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-2, No. 2, Mar 1980.

15. [SHA81] Shapiro, L.G., Moriarty, J.D., Haralick, R.M., Mulgaonkar, P.G., "Matching Three-Dimensional Object Models", Proceedings of the IEEE Conference on Pattern Recognition and Image Processing, 1981.

16. [VOE77] Voelcker, H.B., Requicha, A.A.G., "Geometric Modelling of Mechanical Parts and Processes", COMPUTER, Vol 10, No. 12, Dec 1977.

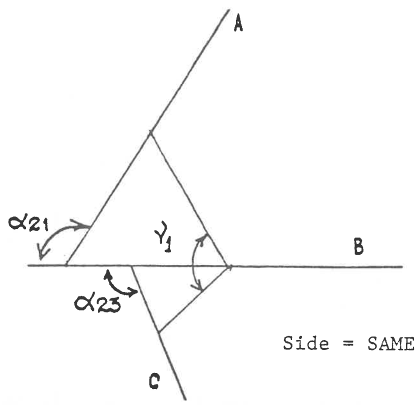

Figure 1: Illustrates three sticks, A, B, and C, participating in the TRIPLES relation. Angle γ_1 is the angle stored with the triple (A,B,C). Angles $\alpha 21$ and $\alpha 23$ are stored with the pairs (A,B) and (B,C), respectively, in the CONNECTS/SUPPORTS relation.

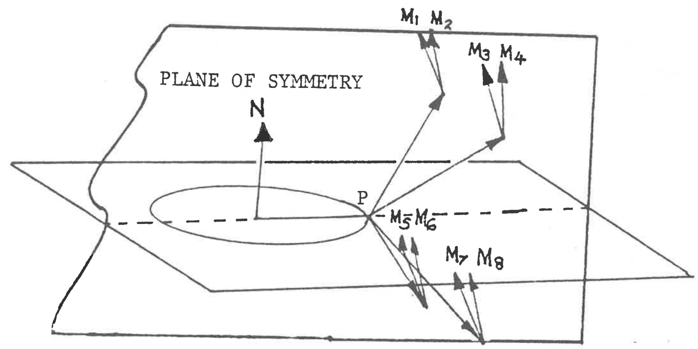

Figure 3: Illustrates the eight different physical interpretations of a logical edge-edge description of two plates specified by three angles. Each vector M_i indicates a different orientation of the second plate relative to the given one.

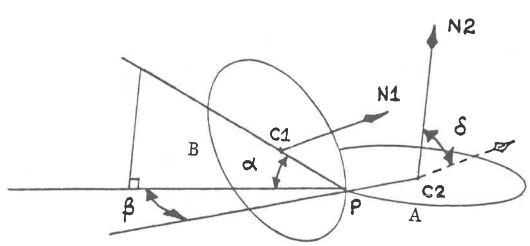

Figure 2: Illustrates an edge-edge binary connection between two plates. C_1 and C_2 are the centers of the plates N_1 and N_2 are the normals to the plates at their centers. Angles alpha, beta and delta are the angles stored in the CONNECTS/SUPPORTS relation.

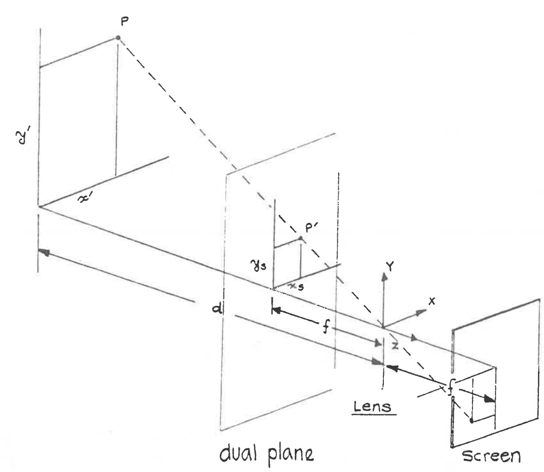

Figure 4: Illustrates the projection of point P in three-dimensional space onto a camera screen behind the lens and dual plane in front of the lens.

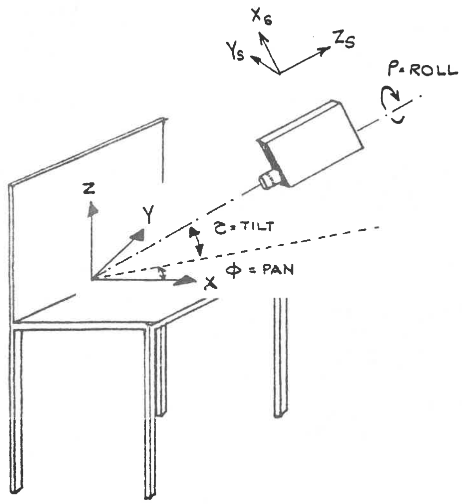

Figure 5: Illustrates the tilt and pan angles of the camera--the two unknown parameters of the perspective normal projection. We assume that the roll angle of the camera is fixed.

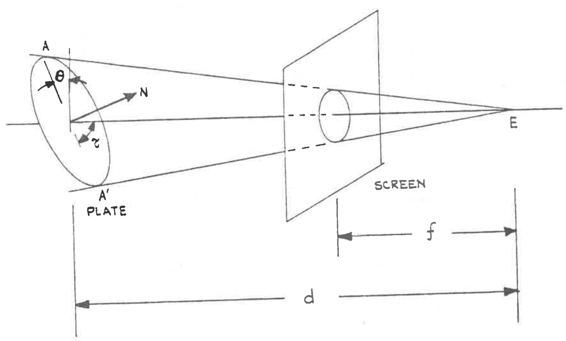

Figure 6: Illustrates the perspective normal projection of a plate (circle) in three-space.

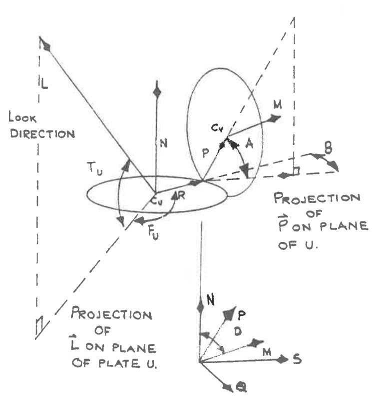

Figure 7: Illustrates edge-edge connections of two plates U and V showing all relevant vectors and angles.

Cv = Center of plate V, Cu Center of Plate U,
N = Normal to plate U, M = Normal to plate V,
R = Vector from Cu to the point of contact,
P = vector from the point of contact to Cv.

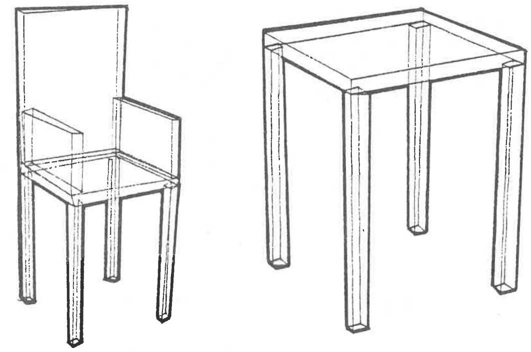

Figure 8: Perspective view of a chair and a table (Sillhouette shown in bold lines).

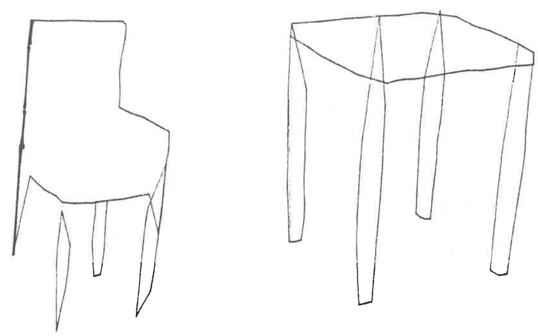

Figure 9: Graph theoretic decomposition of the sillhouettes in Figure 8.

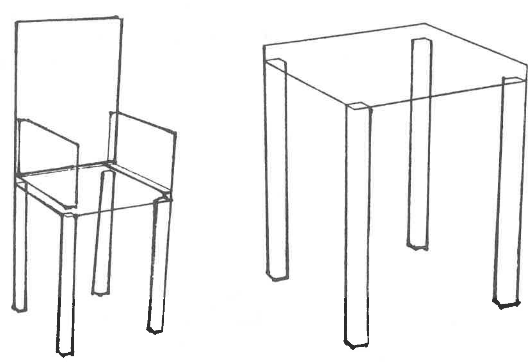

Figure 10: Ideal decomposition of the objects in Figure 8.