Recognition and positioning of partially occluded 3-D objects

Kumar S. Ray and D. Dutta Majumder

Electronics and Communication Sciences Unit, Indian Statistical Institute, 203 B.T. Road, Calcutta-700035, India

Received December 1989
Revised 10 August 1990

Abstract

Ray, K.S. and D. Dutta Majumder, Recognition and positioning of partially occluded 3-D objects, Pattern Recognition Letters 12 (1991) 93-108.

The task of recognizing and positioning the partially occluded three-dimensional (3-D) rigid objects of a given scene is considered. The surfaces of 3-D objects may be planar or curved. The 3-D surface informations are captured through range data (depth map). For recognition we use the principal curvatures, mean curvature and Gaussian curvature as the local descriptions of the surfaces. These curvatures are the local invariant features of the surfaces. A computer vision scheme, based upon the matching between the local features of the 3-D objects in a scene and those of the models which are considered as knowledge data base, is described. Finally, the hypothesis generation and verification scheme is considered for best possible recognition.

Keywords. Computer vision, depth map, 3-D object recognition, occlusion, minimal spanning tree, hypothesis generation and verification, compatible structure, differential geometry.

1. Introduction

A three-dimensional (3-D) vision system deals with images of a 3-D scene. It recognizes 3-D objects and their relative locations in the scene. This capability is quite useful for automatic assembly or industrial inspection. The most important aspect of 3-D object recognition is to establish an efficient algorithm applicable to various scenes. We deal with scenes which have the following characteristics:

(i) Objects are placed at any 3-D position with arbitrary orientation.

(ii) Objects may be stacked and partially occluded.

(iii) Objects have planar and/or smooth curved surfaces.

The existing 3-D object recognition studies mostly deal with two types of input data, namely, the gray-scale intensity image and range image (depth map). At the time we use sensor data to yield information about the real world, it is important to understand the image-formation process. This process has been studied in detail by both computer vision and computer graphics researchers. At each point in an intensity image, the brightness value encodes information about surface geometry (shape, orientation and location), surface reflectance characteristics, surface texture, scene illumination, the distance from the camera to

The authors are also affiliated with the nodal centre for knowledge based computing (KBCS) at Indian Statistical Institute, 203 B.T. Road, Calcutta-700035, India.

an object surface, the characteristics of the intervening medium and the camera characteristics (which include spatial resolution, noise parameters, dynamic range, brightness resolution and lens parameters). Over the years, increased understanding of intensity-image formation (Horn, 1977) and of the constraints of the physical world has led to significant computer vision research developments, including shape from binocular stereo (Grimson, 1980), shape from motion (Jain, 1983; Ullman, 1979), shape from shading (Ikeuchi and Horn, 1981), shape from photometric stereo (Coleman and Jain, 1982; Woodham 1981), shape from texture (Witkin, 1981) and shape from contours (Kanade, 1981). This group of methods is referred to as the shape-from-X techniques. These developments are directed toward the goal of correctly inferring the 3-D structure of a scene from brightness values alone. The great difficulty in reaching that goal is certainly related to the large number of factors encoded in each brightness value during the intensity-image formation process. Range-image formation is conceptually a simpler process than intensity-image formation. At each pixel in a range image, the depth value encodes information about surface geometry and viewing geometry in terms of the distance from the sensor to the object surface, and the range-finder characteristics which include spatial resolution, range resolution, dynamic range, noise parameters and other range-finder parameters that depend on the type of range finder used. One important difference is that scene illumination and surface reflectance are not directly encoded in range values, even though they can definitely affect the accuracy of measured values. Moreover, range finders directly produce the depth (shape) information that the shape-from-X techniques seek to produce. Depth maps are currently receiving a great deal of attention and are very useful for 3-D object recognition. The available literature in this direction includes (Nevatia and Bindord, 1977), (Kuan and Drazovich, 1984), (Smith and Kanade, 1984), (Gennery, 1979), (Bayter and Aggarwal, 1984), (Bhanu, 1982, 1984), (Horaud and Bolles, 1984), (Oshima and Shirai, 1981, 1983), (Faugeras, 1984), (Besl and Jain, 1984) etc.

In this paper we develop a computer vision system to recognize and locate partially occluded 3-D objects present in a range image. The basic concept, to accomplish this task, is based upon the matching of local descriptions of the scene and the models. For ultimate recognition and positioning, the vision system considers a hypotheses generation and verification scheme coupled with an appropriate estimation of the model-to-scene transformations. The control strategy for matching is important particularly in processing a complex scene, since it is very time consuming to match each part of a scene against every possible model. To reduce the number of trials, we first try to get some invariant representative local features of the scene and then find models which have these local features so that this initial matching may guide further processing.

2. Outline of the vision scheme

In the first phase the vision system builds up the model data base of the objects which are most likely to appear in the scene. If one view is not enough to describe an object, several typical views are shown to the vision system and multiple models of an object are made. When we talk about the models of a 3-D object, we mean the local invariant representative features of the surface of the 3-D objects. These features will essentially construct the models of the 3-D objects. For instance, we consider the principal curvatures, mean curvature and Gaussian curvature to characterize the surface of a 3-D object.

In the second phase the vision system makes a description of the unknown scene in the same way as in the first phase. That means, again the principal curvatures, mean curvature and Gaussian curvature are used to characterize the surface of the 3-D scene.

The model and scene descriptions are given in Section 4.

The third phase is the recognition and positioning phase. At the initial stage of recognition, the scene descriptions (principal curvatures of the scene edge-points) are matched with the models' descriptions (principal curvatures of the model edge-points) so that the edge-points of the model objects are found sequentially in the scene. The detail description of this initial matching is described in Section 5.1.

After the task of initial recognition is over, appropriate coordinate transformations are performed. This is discussed in Sections 5.1 and 5.2.

Based upon the initial match and coordinate transformations, the vision system initially hypothesizes the locations of the edge-points of the model objects in the scene. This initial hypothesis, which uses the verifications of structural compatibility of the edge-points, is discussed in detail in Section 5.3.

Now the surfaces of the model objects are matched with those of the scene. This is mainly performed through the matching of the non-edge-points of the model object with those of the scene. This task is described in Section 5.4.

In Section 5.5 the vision system finally hypothesizes the locations of the 3-D objects in the scene.

Finally the vision system verifies the hypotheses. This verification process is discussed in Section 5.6.

This process of hypothesizing and verifying the locations of 3-D objects continues until everything in the scene is understood or no more matches can be found.

3. Mathematical preliminaries

We have seen (Besl and Jain, 1984; Ray and Dutta Majumder, 1989) that curvature, torsion and speed uniquely determine the shape of space curves. These characteristics are the ideal type of characteristic for a mathematical entity. They are invariant to coordinate transformations and they have a one-to-one relationship with curve shapes. We now briefly discuss similar surface characteristics by generalizing the mathematical frame work of curves to surfaces. Surfaces are the natural geometric generalization of curves. For detailed discussions on surface characteristics interested readers are referred to (Hsiung, 1981; Besl and Jain, 1984).

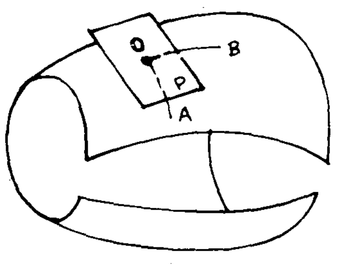


Figure 1. Curvature of curves on a surface.

3.1. Curvature of curves on a surface and the approximate surface area

The curvature of a surface at a given point is characterized by the rate at which the surface leaves its tangent plane. But in different directions the surface may leave its tangent plane at different rates. For example the surface illustrated in Figure 1 leaves the plane P in the direction OA at a faster rate than in the direction OB. So it is natural to define the curvature of a surface at a given point by means of the whole set of curvatures of curves lying in the surface and passing through the given point in different directions.

Thus it appears that a surface may be curved arbitrarily in many directions. In fact this is not so. In differential geometry (DoCarmo, 1976), it is shown that at each point of a surface there exist two particular directions such that:

(i) they are mutually perpendicular;

(ii) the curvatures K_1 and K_2 of the normal sections in these directions, as shown in Figure 2, are the smallest and largest values of the curvatures of all normal sections;

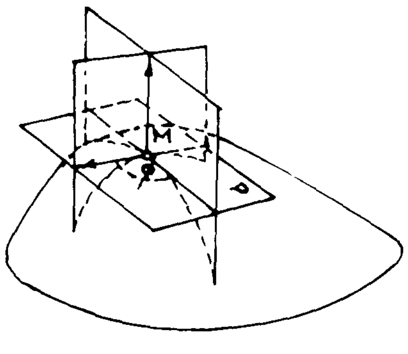


Figure 2. Principal directions and principal curvatures of the surface at a point.

(iii) the curvature $K(\phi)$ of the normal section rotated from the section with curvature K_1 by the angle ϕ is expressed by the formula

$$K(\phi) = K_1 \cos^2 \phi + K_2 \sin^2 \phi. \tag{1}$$

Such directions are called the principal directions and the curvatures K_1 and K_2 are called the principal curvatures of the surface at the given point.

Thus, if we know K_1 and K_2, the curvature of any curve in the surface is defined by the direction of its tangent and the angle between its osculating plane and the normal to the surface. Consequently, the character of the curvature of a surface at a given point is defined by the two numbers K_1 and K_2. Their absolute values are equal to the curvatures of two mutually perpendicular normal sections and their signs show the direction of the concavity of the respective normal sections with respect to a chosen direction on the normal.

In many questions of the theory of surfaces the most important role is played, not by the principal curvatures themselves, but by certain quantities dependent on them, namely the mean curvature and the Gaussian curvature of the surface at a given point.

The mean curvature H of a surface at a given point is the average of the principal curvatures

$$H = \tfrac{1}{2}(K_1 + K_2). \tag{2}$$

The Gaussian curvature of a surface at a given point is the product of the principal curvatures:

$$K = K_1 K_2. \tag{3}$$

The Gaussian curvature is an intrinsic property of the surface, since it depends only on the coefficients of the first fundamental form and their derivatives, whereas the two principal curvatures and the mean curvature are extrinsic properties. The coefficients of the first and second fundamental forms, i.e., E, F, G, e, f, g are given below. They determine the surface uniquely up to a rigid body transformation (DoCarmo, 1976). The principal curvatures at a point on a surface are computed in terms of the parameters u and v as follows. Let $X(u, v)$ represent the surface.

$$X(u, v) = (x(u, v), y(u, v), z(u, v)), \tag{4}$$

$$X_u = \frac{\partial X}{\partial u} = (x_u, y_u, z_u), \tag{5}$$

$$X_v = \frac{\partial X}{\partial v} = (x_v, y_v, z_v), \tag{6}$$

$$X_{uv} = \frac{\partial^2 X}{\partial u \partial v} = (x_{uv}, y_{uv}, z_{uv}), \tag{7}$$

$$X_{uu} = \frac{\partial^2 X}{\partial u^2} = (x_{uu}, y_{uu}, z_{uu}), \tag{8}$$

$$X_{vv} = \frac{\partial^2 X}{\partial v^2} = (x_{vv}, y_{vv}, z_{vv}). \tag{9}$$

Let

$$E = |X_u|^2 = (x_u^2 + y_u^2 + z_u^2), \tag{10}$$

$$F = X_u \cdot X_v = (x_u x_v + y_u y_v + z_u z_v), \tag{11}$$

$$G = |X_v|^2 = (x_v^2 + y_v^2 + z_v^2). \tag{12}$$

The unit normal is then given by

$$N = \frac{X_u \times X_v}{\sqrt{EG - F^2}}. \tag{13}$$

The Gaussian curvature K at any point on a surface is defined as the product of the two principal curvatures K_1 and K_2 and is further represented by

$$K = \frac{eg - f^2}{EG - F^2} \tag{14}$$

where e, f and g are given by

$$e = N \cdot X_{uu}, \tag{15}$$

$$f = N \cdot X_{uv}, \tag{16}$$

$$g = N \cdot X_{vv}. \tag{17}$$

The mean curvature H is given by

$$H = \frac{eG - 2fF + gE}{2(EG - F^2)}, \tag{18}$$

and the principal curvatures are

$$K_1 = H - \sqrt{H^2 - K}, \tag{19}$$

$$K_2 = H + \sqrt{H^2 - K}. \tag{20}$$

For further details on the computations of surface curvatures interested readers are referred to (Besl and Jain, 1984, 1988). Based upon the above computations we can classify surface shapes, as shown in Figure 3, into eight basic categories.

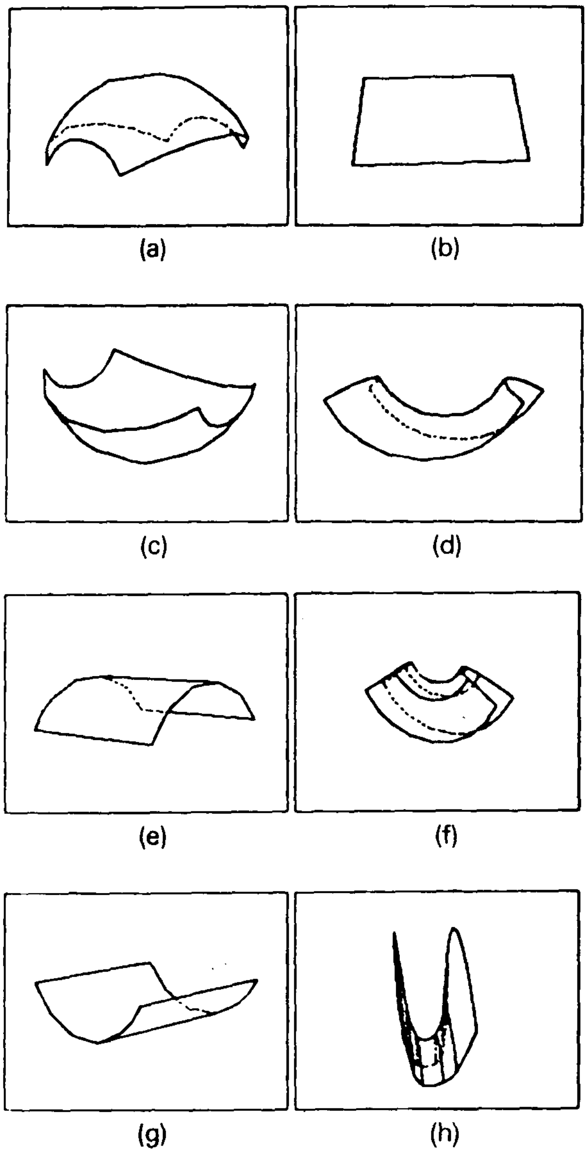


Figure 3. Eight fundamental surface shapes: (a) peak surface $H<0$, $K>0$; (b) flat surface: $H=0$, $K=0$; (c) pit surface: $H>0$, $K>0$; (d) minimal surface: $H=0$, $K<0$; (e) ridge surface: $H<0$, $K=0$; (f) saddle ridge: $H<0$, $K<0$; (g) valley surface: $H>0$, $K=0$; (h) saddle valley: $H>0$; $K<0$.

To obtain a formula for surface area, we consider a curvilinear rectangle bounded by the coordinate curves $u=u_0$, $v=v_0$, $u=u_0+\Delta u$, $v=v_0+\Delta v$ and we take as an approximation to it the parallelogram lying in the tangent plane and bounded by the vectors $X_u\Delta u$, $X_v\Delta v$, tangent to the coordinate curves (Figure 4). The area of this parallelogram is

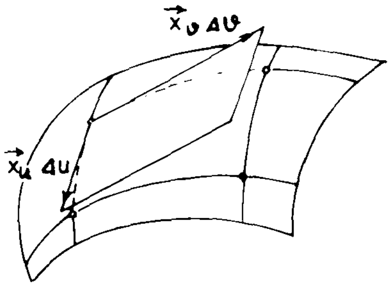


Figure 4. The parallelogram lying in the tangent plane and bounded by the vectors $X_u\Delta u$, $X_v\Delta v$.

$$\Delta S = |X_u|\,|X_v|\,\Delta u\,\Delta v \sin\psi,$$

where ψ is the angle between X_u and X_v. Since $\sin\psi=\sqrt{1-\cos^2\psi}$, it follows that

$$\Delta S = |X_u|\,|X_v|\,\Delta u\,\Delta v\sqrt{1\text{-}\cos^2\psi}$$
$$= \sqrt{|X_u|^2\,|X_v|^2 - |X_u|^2|X_v|^2\cos^2\psi}\;\Delta u\,\Delta v.$$

Recall that $|X_u|^2=E$, $|X_v|^2=G$ and $|X_u|\,|X_v|\cos\psi = X_u\cdot X_v = F$. Hence, we get

$$\Delta S = \sqrt{EG-F^2}\,\Delta u\,\Delta v. \tag{21}$$

Summing up the areas of the parallelograms over the depth map regions we get the approximate surface area of the region (Besl and Jain, 1986). The continuous domain formulation of equation (21) is

$$\text{Surface area} = \iint_D \sqrt{EG-F^2}\,\mathrm{d}u\,\mathrm{d}v,$$

where summation is replaced by integration taken over the domain D of the variables u and v, in the limit Δu, $\Delta v\to 0$.

4. Computations of model and scene descriptions

Based upon the expressions of Section 3 we compute the principal curvatures (K_1, K_2), mean curvature (H) and Gaussian curvature (K) at different points of the model objects and scene. These curvatures are the local features of the model objects and scene. After the computations of these curvatures we consider the following.

(i) We extract the edge-points of the model objects and scene. In the presence of an edge the principal curvatures will achieve a local maximum (Vemuri et al., 1987). Therefore it is appropriate to declare as edge-points those points at which the principal curvatures are above the given thresholds (Cline, 1981). Edge-points are further clarified in Figure 5. The edge-points of the models and scene will be used for initial matching.

(ii) We classify each non-edge-point into one of the eight categories shown in Figure 3. Finally we group all neighbouring points of the same type in a patch and similar neighbouring patches into a region (Oshima and Shirai, 1979, 1983). Each patch (or region) of the models and scene is described by a representative point. In case of patches we denote such points by P_i, $i = 1, 2, 3, \ldots$, and in case of regions we denote such points by R_i, $i = 1, 2, 3, \ldots$. Each P_i (R_i) is associated with an appropriate three-dimensional coordinate and the values of the mean curvature and Gaussian curvature of the patch (region). These non-edge-points will be used for surface matching. For further clarity see Figure 5. Non-edge-points are always surrounded by contours or edge-points of the object.

5. Recognition and positioning of 3-D objects

The task of recognition and positioning of partially occluded 3-D objects is performed sequentially as follows.

5.1. Initial matching between model edge-points and scene edge-points

To match the edge-points we use the Euclidean distance between the principal curvatures of the model edge-point and scene edge-point. An Euclidean distance J_1 greater than a threshold T_{J_1} indicates two dissimilar edge-points and less than or equal to T_{J_1} results in two similar edge-points. This decision rule decides if a scene edge-point matches a model edge-point (see the Appendix). After finding the match we compute the appropriate 3-D coordinate transformation T to place the coordinates of the matched edge-points of the models on the corresponding edge-points of the scene. T is represented by the rotational parameters α (rotation around the x-axis), β (rotation around the y-axis), γ (rotation around the z-axis) and the translation parameters t_x (translation in the x-direction), t_y (translation in the y-direction), t_z (translation in the z-direction).

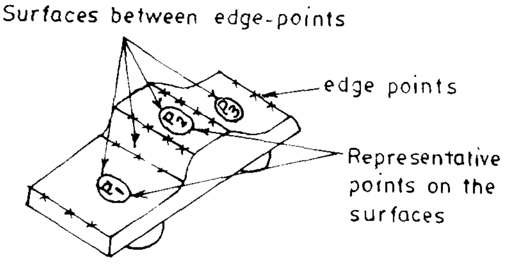

Surfaces between edge-points
edge points
R1
R2
R3
Representative points on the surfaces

Figure 5. 3-D car representing the edge-points and the 3-D surface orientation.

The model edge-points form a dictionary of the model features which is used as knowledge data base. Considering the scene edge-points the matching algorithm looks up the possible matches to the model edge-points with the help of this dictionary.

5.2. Coordinate transformations of the initially matched points

After achieving the initially matched edge-points between model objects and scene, our aim is to estimate the three-dimensional coordinate transformation T which consists of rotation (rotation about three axes) and translation (translations in the directions of three axes) and which transforms the model objects into the scene reference frame. The transformation T is applied only on the matched edge points of the model objects. To achieve the transformation T we apply the concept of composite transformation through matrix multiplication or concatenation (Plastock and Kalley, 1986; Ballard and Brown, 1982). Actually, to estimate the rotational and translational parameters of the edge-points of the model objects into the scene reference frame we solve a set of nonlinear equtions using the iterative method of Newton–Raphson (Carnaham et al., 1969).

5.3. Generation of initial hypothesis

The initial hypothesis makes a list of scene edge-points and their matches for each possible model that could appear in the scene. To hypothesize an edge-point of the model object in the scene, mutual compatibility constraints extract a set of consistent matches from a model's match list. First we discuss the mutual match compatibility between two matches. Then we discuss a group of mutually compatible matches.

5.3.1. *Mutual compatibility and multiple matches*

Suppose that two matches Θ_i $(i=1,2)$ contain a scene edge-point S_i and a model 'edge-point M_i. The following four rules test Θ_1 and Θ_2 for their mutual compatibility.

(i) $S_1 \neq S_2$.

(ii) $M_1 \neq M_2$.

(iii) The two scene edge-points refer to model edge-points belonging to the same model (M_1 and M_2 belong to the same model).

(iv) The structure between the two scene features equals the structure between the two model features.

The first three rules are self-explanatory. The fourth rule comes from the assumption that rigid objects make up the model set. To have the same structure, two matches should have the same rotation and translation parameters. The very unlikely possibility that two matches have exactly the same coordinate transform parameters requires that the distance between them to be less than some small value ε. That is

$$\begin{aligned} \delta_\alpha &= |\alpha_1 - \alpha_2| < \varepsilon_\alpha, \\ \delta_\beta &= |\beta_1 - \beta_2| < \varepsilon_\beta, \\ \delta_\gamma &= |\gamma_1 - \gamma_2| < \varepsilon_\gamma \end{aligned} \tag{22}$$

and

$$\begin{aligned} \delta_x &= |t_{x_1} - t_{x_2}| < \varepsilon_t, \\ \delta_y &= |t_{y_1} - t_{y_2}| < \varepsilon_t, \\ \delta_z &= |t_{z_1} - t_{z_2}| < \varepsilon_t. \end{aligned} \tag{23}$$

To compare two matches, we compare the rotational parameters first. If the difference between the rotational parameters is less than a certain threshold value as indicated by equation (22), then we fix the rotation parameter to the same value for both matches. After this we compute the translational parameters. If the difference between the translational parameters has a value less than some threshold value as indicated by equation (23), then the matches have compatible coordinate structure (Ray and Dutta Majumder, 1989; Koch and Kashyap, 1985). Otherwise the matches do not meet the structural compatibility requirements.

Suppose a group of compatible model features represents a high-level model feature which agrees with a high-level scene feature or a large portion of the model identifiable in the scene. There are already a lot of schemes (Ballard, 1983; Hakalathi, 1984; Stockman et al., 1982; Bolles and Cain, 1982; Grimson and Lozano-Pérez, 1984; Fischer and Bolles, 1981; Koch and Kashyap, 1985) available for multiple compatible matches. However, we apply the minimal spanning tree algorithm (Shamos and Hoey, 1975). This algorithm has several benefits which are briefly discussed below. The algorithm starts with the smallest amount of data needed to form a solution and then adds consistent data instead of using all the data to form a cluster around the match. The compatible matches satisfy the four compatibility conditions mentioned earlier. Conflicts can occur between pairs of matches which can be solved by taking the match with the highest structural compatibility. Thus, we find out the largest set of mutually compatible matches. The complexity of the algorithm is of the order of $O(n \log n)$ where n represents the number of points to be clustered. This technique can avoid stray matches by fixing the cluster centers (see the Appendix). An example of our matching procedure for finding the largest compatible match is discussed in the Appendix.

The storage requirements to realize this algorithm are not large. We have to store the values of the principal curvatures (K_1, K_2) and of the coordinates at different edge-points of the model objects and scene.

At the end of the initial hypothesis we complete some tasks for non-edge-point matching in the following way.

We record the coordinates, mean curvatures and Gaussian curvatures of the representative points of the patches (regions) of the model objects and scene. These non-edge-points are surrounded by the matched edge-points of the model objects and scene. For further clarity see Figure 5.

5.4. *Matching of the non-edge-points of the model objects and scene*

After matching the edge-points, we try to match the patches (regions) which are surrounded by the matched edge-points of model objects and scene and which are formed by a similar type of non-edge-points mentioned in item (ii) of Section 4. At the time of matching the patches (regions), we consider the representative point (see Figure 5) of each patch (region) of the model objects and scene. We apply the appropriate transformation T, which is already estimated in Section 5.1, to each representative point of the patch (P_i) or region (R_i) of the model objects and find the corresponding representative point of the patch (region) of the scene. At the time we match the patch (P_i) or region (R_i) of the model objects with that of the scene we assume the similar four compatibility conditions mentioned in Section 5.3.1. To maintain the same compatibility structure, the two distinct representative points of patches (regions) of model objects and scene should satisfy the inequality type constraints of equations (22) and (23). We compare the transformed coordinates of the representative points of patches (regions) of the model objects with the coordinates of the corresponding points in the scene. If the compared results (comparison is performed through Euclidean measure) are below a certain threshold value, then the transformation is acceptable. Otherwise it is not acceptable. Finally, we compare the mean curvature and Gaussian curvature of the representative points of patches (regions) of model objects with those of the scene. These curvatures essentially characterize the orientation of the surface of the patches (regions). Different surface orientations are shown in Figure 3. If the compared values (comparison is performed through Euclidean measure) are below a certain threshold, then it indicates similarity between two patches (regions). Thus the matching of the non-edge-points indicates similarity in orientation of surfaces which exist between different edge-points of the model objects and scene.

5.5. *Generation of the final hypothesis*

Finally we hypothesize the locations of particular views of the model objects in the three-dimensional scene. Thus, we construct the three-dimensional scene using the hypothesized model objects. In the next section we try to verify how far the hypotheses are correct.

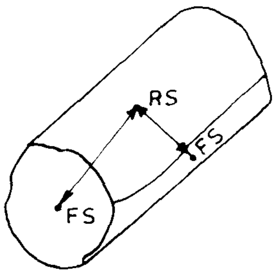
5.6. *Verification of the hypotheses*

The verification of the hypotheses can be performed through the verification of the following three items.

(i) Verify the total number of patches (regions) of the original 3-D scene with that of patches (regions) of the scene formed by the hypothesized model objects. If the hypotheses are correct, then the total number is the same in both cases.

(ii) Compare the adjacency relation of patches (regions) of the original 3-D scene with that of patches (regions) of the scene formed by the hypothesized model objects. This actually reduces to a graph matching problem mentioned in (Oshima and Shirai, 1983). The nodes of the graph represent the type of the patches (regions) and an arc between two nodes represents the adjacency relation between the two patches (regions). For further clarity see Figure 6.

(iii) Finally, using equation (21), we compute the approximate surface area of the original 3-D scene and the approximate surface area of the scene formed by the hypothesized model objects. If the difference between the two approximately computed surface areas (Besl and Jain, 1986) is below a certain threshold, then the hypotheses are consistent.



RS = Ridge surface with $H<0$, $K=0$
FS = Flat surface with $H=0$, $K=0$

Figure 6. Adjacency relation between patches/regions.

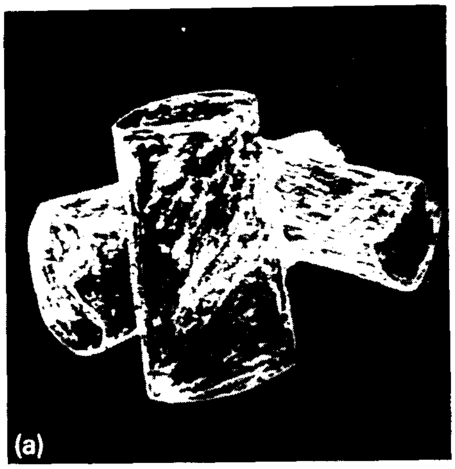


Figure 7(a). Original 3-D scene.

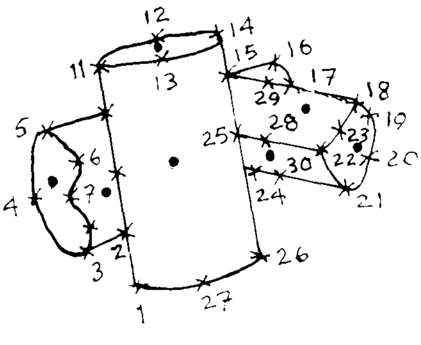


Figure 7(b). Edge-points and non-edge-points of the 3-D scene

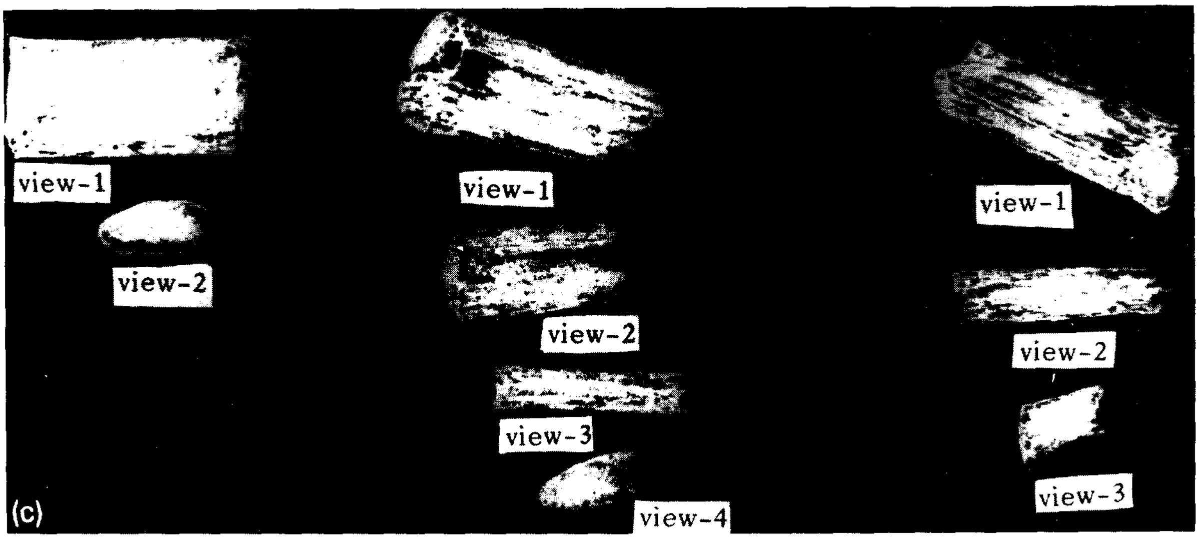


Figure 7(c). Different views of the model objects.

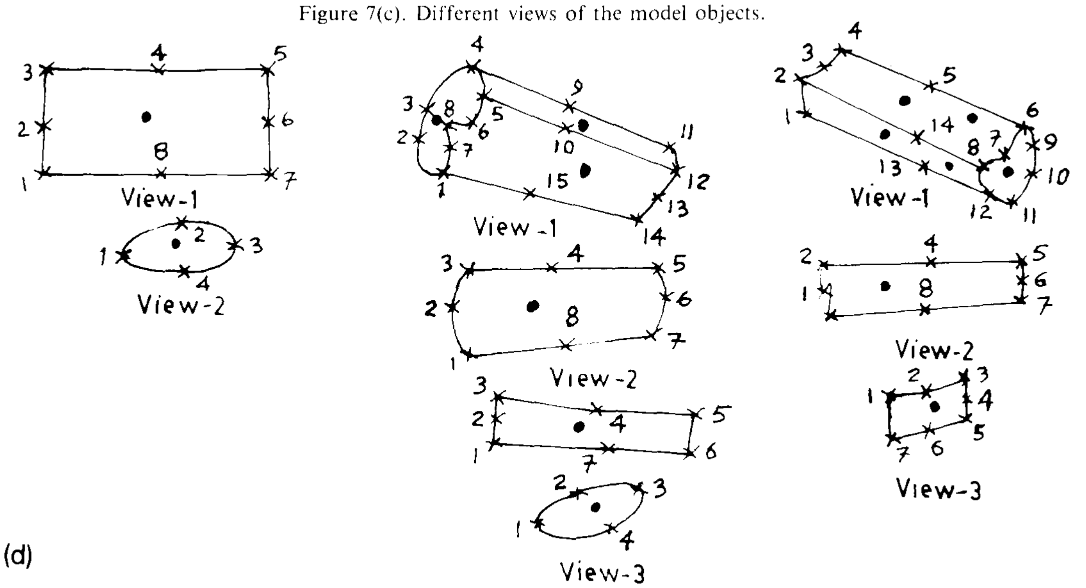
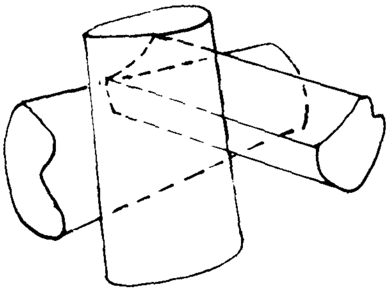


Figure 7(d). Edge-points and non-edge points of the model objects.



(e)

Figure 7(e). Hypothesized 3-D scene.

6. Experimental results and discussion

Case study 1

Figure 7(a) represents a 3-D scene formed by wooden blocks. The stars of Figure 7(b) represent the edge-points (see item (i) of Section 4) of the real 3-D scene shown in Figure 7(a). Figure 7(c) represents different views of the model objects. The stars of Figure 7(d) represent the edge-points of the model objects. An Euclidean measure J_1 and an appropriate threshold T_{J_1} determine if two edge-points match and result in a 3-D coordinate transform for the ultimate matching through structural compatibility using the minimal spanning tree algorithm. Tables 1, 2 and 3 represent the matches that make up compatible structures for the model object-1, model object-2 and model object-3.

The following coordinate transformations give the locations of the best possible hypotheses for the edge-points.

For model object-1

$$\begin{aligned} \text{view-1:}\quad & \alpha'=153,\ \beta'=89,\ \gamma'=212,\\ & T_x=-141,\ T_y=231,\ T_z=122\\ \text{view-2:}\quad & \alpha'=17,\ \beta'=10,\ \gamma'=5,\\ & T_x=51,\ T_y=89,\ T_z=453 \end{aligned} \tag{24}$$

For model object-2

$$\begin{aligned} \text{view-1:}\quad & \alpha'=164,\ \beta'=318,\ \gamma'=118,\\ & T_x=121,\ T_y=51,\ T_z=432 \end{aligned} \tag{25}$$

For model object-3

$$\begin{aligned} \text{view-1:}\quad & \alpha'=13,\ \beta'=04,\ \gamma'=13,\\ & T_x=131,\ T_y=332,\ T_z=222 \end{aligned} \tag{26}$$

The dots of Figure 7(b) represent the selective

Table 1

Cluster (using minimal spanning tree) for determining the edge-points of the model object-1 (see Figure 7(d)) in the scene represented by Figure 7(b)

Cluster of compatible matches for edge-points of the model object-1

		J_1 between													
Scene	View-1	K_1	K_2	α	β	γ	t_x	t_y	t_z	α'	β'	γ'	t'_x	t'_y	t'_z
1	3	0.0012	0.004	150	90	210	−145	230	120	153	89	212	−141	231	121
9	4	0.0011	0.0005	154	87	215	−142	232	124	153	89	212	−142	231	122
25	8	0.0015	0.0025	155	88	212	−140	234	122	153	89	212	−141	232	121
26	1	0.0045	0.0015	152	91	213	−144	233	126	153	89	212	−140	230	123
27	2	0.0033	0.0025	153	89	209	−140	231	128	153	89	212	−141	231	122

$$\alpha'=\frac{\sum\alpha}{n}=153,\quad \beta'=\frac{\sum\beta}{n}=89,\quad \gamma'=\frac{\sum\gamma}{n}=212,\quad T_x=\frac{\sum t'_x}{n}=-141,\quad T_y=\frac{\sum t'_y}{n}=231,\quad T_z=\frac{\sum t'_z}{n}=122$$

		J_1 between													
Scene	View-2	K_1	K_2	α	β	γ	t_x	t_y	t_z	α'	β'	γ'	t'_x	t'_y	t'_z
11	1	0.0023	0.0014	15	10	05	53	95	450	17	10	05	50	90	452
12	2	0.0018	0.0016	18	8	03	55	90	451	17	10	05	51	89	453
13	4	0.0022	0.0022	19	9	04	52	91	455	17	10	05	50	88	454
14	3	0.0011	0.0034	16	12	06	51	93	457	17	10	05	51	89	452

$$\alpha'=\frac{\sum\alpha}{n}=17,\quad \beta'=\frac{\sum\beta}{n}=10,\quad \gamma'=\frac{\sum\gamma}{n}=5,\quad T_x=\frac{\sum t'_x}{n}=51,\quad T_y=\frac{\sum t'_y}{n}=89,\quad T_z=\frac{\sum t'_z}{n}=453$$

Table 2
Cluster (using minimal spanning tree) for determining the edge-points of the model object-2 (see Figure 7(d)) in the scene represented by Figure 7(b)

Cluster of compatible matches for edge-points of the model object-2

		J_1 between													
Scene	View-1	K_1	K_2	α	β	γ	t_x	t_y	t_z	α'	β'	γ'	t'_x	t'_y	t'_z
3	1	0.0011	0.0015	165	318	118	120	50	430	164	318	118	121	51	431
4	3	0.0020	0.0012	160	315	115	123	51	432	164	318	118	122	52	432
5	4	0.0035	0.0011	163	318	116	124	55	433	164	318	118	123	53	431
6	5	0.0014	0.0012	167	319	120	120	54	434	164	318	118	121	52	433
7	8	0.0019	0.0025	168	320	120	121	53	435	164	318	118	122	50	431
8	7	0.0014	0.0014	160	322	118	125	52	431	164	318	118	120	51	433
16	11	0.0025	0.0035	163	314	119	126	50	434	164	318	118	121	52	432
17	12	0.0012	0.0011	164	315	128	122	51	432	164	318	118	120	50	431

$$\alpha' = \frac{\sum \alpha}{n} = 164, \quad \beta' = \frac{\sum \beta}{n} = 318, \quad \gamma' = \frac{\sum \gamma}{n} = 118, \quad T_x = \frac{\sum t'_x}{n} = 121, \quad T_y = \frac{\sum t'_y}{n} = 51, \quad T_z = \frac{\sum t'_z}{n} = 432$$

Table 3
Cluster (using minimal spanning tree) for determining the edge-points of the model object-3 (see Figure 7(d)) in the scene represented by Figure 7(b)

Cluster of compatible matches for edge-points of the model object-3

		J_1 between													
Scene	View-1	K_1	K_2	α	β	γ	t_x	t_y	t_z	α'	β'	γ'	t'_x	t'_y	t'_z
18	6	0.0041	0.0024	10	05	12	135	331	223	13	04	13	131	332	220
19	9	0.0011	0.0021	15	04	13	136	335	220	13	04	13	130	331	221
20	10	0.0014	0.0035	16	03	14	131	330	224	13	04	13	131	333	222
21	11	0.0051	0.0044	12	06	15	132	336	225	13	04	13	132	331	221
22	8	0.0011	0.0013	14	02	10	133	332	221	13	04	13	131	332	222
23	7	0.0015	0.0019	15	04	15	135	331	226	13	04	13	131	331	223
28	14	0.0021	0.0027	12	03	10	131	334	224	13	04	13	132	332	221
29	5	0.0032	0.0012	13	02	12	136	336	225	13	04	13	130	333	222
30	13	0.0004	0.0010	14	05	13	132	335	222	13	04	13	131	331	223

$$\alpha' = \frac{\sum \alpha}{n} = 13, \quad \beta' = \frac{\sum \beta}{n} = 04, \quad \gamma' = \frac{\sum \gamma}{n} = 13, \quad T_x = \frac{\sum t'_x}{n} = 131, \quad T_y = \frac{\sum t'_y}{n} = 332, \quad T_z = \frac{\sum t'_z}{n} = 222$$

non-edge-points of the scene and the dots of Figure 7(d) represent the selective non-edge-points of the model objects (see Section 5.4). The non-edge-points of the model objects experienced the same 3-D coordinate transformation as mentioned for the best possible hypothesis for the edge-points. Checking the structural compatibility, as mentioned in Sections 5.3 and 5.4, we compare the mean curvature and Gaussian curvature of the non-edge-points of the model objects and scene. An Euclidean measure J_2 and an appropriate threshold T_{J_2} determine if two non-edge-points match.

Finally we hypothesize that the 3-D scene of Figure 7(a) is formed by the following combinations.

(i) View-1 and -2 of the model object-1 with the 3-D transformation shown in equation (24).

(ii) View-1 of the model object-2 with the 3-D transformation shown in equation (25).

(iii) View-1 of the model object-3 with the 3-D transformation shown in equation (26).

The hypothesized 3-D scene is shown in Figure 7(e). To verify the hypotheses we check the three items of Section 5.6.

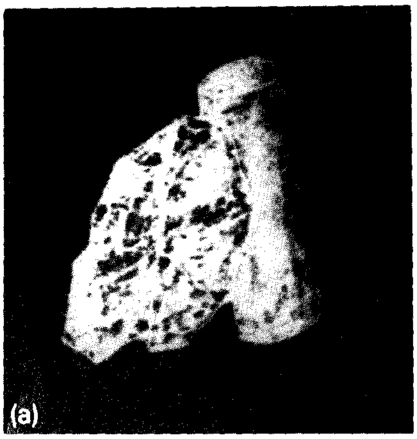


Figure 8(a). Original 3-D scene.

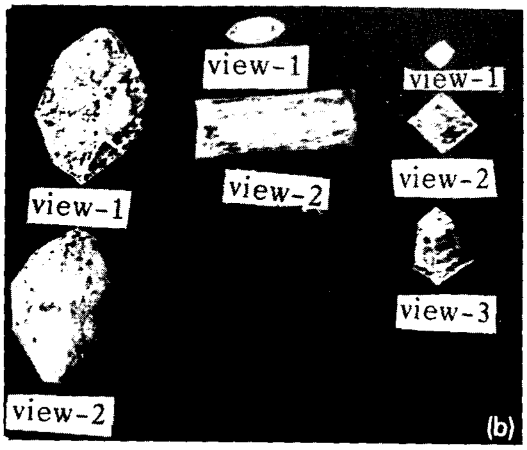


Figure 8(b). Different views of the model object.

Case study 2

In the second experiment we consider the 3-D scene shown in Figure 8(a). Similar to Case study 1 we ultimately hypothesize that the 3-D scene of Figure 8(a) is formed by the following combinations.

(i) View-1 of the model object-1 with the following 3-D transformation:

$$\alpha' = 160,\ \beta' = 135,\ \gamma' = 65,$$
$$T_x = 240,\ T_y = -175,\ T_z = 340.$$

(ii) View-1 and -2 of the model object-2 with the following 3-D transformations:

$$\alpha' = 145,\ \beta' = 75,\ \gamma' = 350,$$
$$T_x = 150,\ T_y = 210,\ T_z = 355$$

and

$$\alpha' = 231,\ \beta' = 145,\ \gamma' = 175,$$
$$T_x = 295,\ T_y = 345,\ T_z = 312.$$

(iii) View-3 of the model object-3 with the following 3-D transformation:

$$\alpha' = 195,\ \beta' = 97,\ \gamma' = 325,$$
$$T_x = 135,\ T_y = 314,\ T_z = 175.$$

Finally the hypotheses are verified.

The choice of thresholds is very important in both cases. If the values of thresholds are set at a higher level, different surfaces and edge-points tend to be recognized as identical. On the other hand identical surfaces and edge-points may be recognized as different if the threshold values are set at lower values because of noise and digitization errors. In the PDP-11 (24) computer the total CPU time (starting from the scene and model descriptions to hypotheses verifications) required for Case study 1 is 5.75 minutes and for Case Study 2 is 6.35 minutes.

7. Conclusion

We have presented a model based vision scheme to recognize and locate partially occluded three-dimensional rigid objects. The input to the vision system is the range data. The system describes a scene in terms of planes, curved surfaces and edge-points. The models of objects are built in the system by showing them one at a time. Sometimes multiple views of an object model is necessary for complete description. Objects in an unknown scene are recognized by matching the description of the scene to those of the models. During the matching process we have used some standard results of differential geometry for curves and surfaces. Based on the matched descriptions of the scene and model objects we form the hypotheses about the 3-D scene. Finally we verify the hypotheses. The validity of the present vision scheme is tested on two examples and very promising results are obtained.

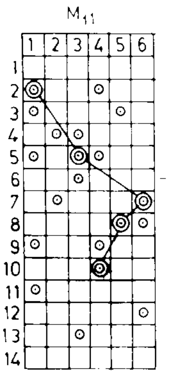
Table 4

Model edge points ⟹ ; Scene edge points ⟹

	Model-1, view-1 M_{11}						Model-1, View-k M_{1K}							Model-n, view 1 M_{n1}					Model-n, view-r M_{nr}								
	1	2	3	4	5	6	1	2	3	4	5	6	7	1	2	3	4	5	1	2	3	4	5	6	7	8	9
1	·	·	·	·	·	·	⊙	·	·	·	○	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
2	⊙	·	·	⊙	·	·	·	⊙	·	·	·	·	·	⊙	·	⊙	·	·	⊙	·	⊙	·	·	·	·	·	·
3	⊙	·	·	·	⊙	·	·	·	⊙	⊙	·	·	⊙	·	·	·	·	·	·	·	·	·	·	·	·	·	·
4	·	⊙	⊙	·	·	·	·	·	·	⊙	·	·	·	·	·	⊙	⊙	⊙	⊙	·	·	·	⊙	·	·	·	·
5	·	·	⊙	⊙	·	·	·	·	⊙	·	·	⊙	·	·	·	·	·	⊙	·	·	·	·	·	·	·	·	·
6	·	·	⊙	·	·	·	⊙	·	·	·	·	⊙	·	·	·	⊙	⊙	·	·	·	·	·	·	·	·	·	·
7	·	⊙	·	·	·	⊙	·	·	·	○	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
8	·	·	·	·	⊙	⊙	·	·	⊙	·	·	·	·	·	⊙	·	·	⊙	·	·	·	·	⊙	·	⊙	·	·
9	⊙	·	·	⊙	·	·	·	⊙	·	·	·	·	⊙	·	·	⊙	·	·	·	·	·	·	·	·	·	·	·
10	·	·	·	⊙	·	·	·	·	⊙	⊙	·	·	·	·	·	·	⊙	·	·	·	·	·	·	·	·	·	·
11	⊙	·	·	·	·	·	·	·	·	·	⊙	·	·	⊙	·	⊙	·	·	⊙	⊙	·	·	·	·	·	·	·
12	·	·	·	·	·	·	⊙	·	·	⊙	·	·	⊙	·	·	·	·	·	·	·	·	·	·	·	·	·	⊙
13	·	·	⊙	·	·	·	·	·	⊙	·	·	·	·	·	·	·	⊙	·	·	·	·	·	·	·	·	·	·
14	·	·	·	·	·	·	·	·	⊙	·	⊙	·	·	·	·	⊙	·	·	·	·	⊙	⊙	·	·	·	·	·

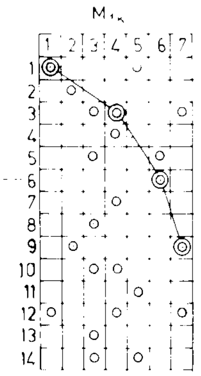
· Indicates the difference between the model-edge points and the scene edge-points
⊙ Indicates the similarity between the model edge-points and scene edge-points

Table 5



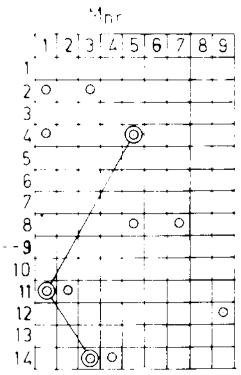
	1	2	3	4	5	6
1						
2	◎			⊙		
3	⊙				⊙	
4		⊙	⊙			
5	⊙		◎	⊙		
6			⊙			
7		⊙				◎
8					◎	⊙
9	⊙			⊙		
10				◎		
11	⊙					
12						⊙
13			⊙			
14						

Table 6



	1	2	3	4	5	6	7
1	◎				○		
2		○					
3			○	◎			○
4				○			
5			○			○	
6						◎	
7				○			
8			○				
9		○					◎
10			○	○			
11					○		
12	○			○			○
13			○				
14			○		○		

Table 7



	1	2	3	4	5	6	7	8	9
1									
2	○		○						
3									
4	○				◎				
5									
6									
7									
8					○		○		
9									
10									
11	◎	○							
12									○
13									
14			◎	○					

◎ ≡ Indicates a node which would consist on match. An arc between two nodes satisfies the four compatibility constraints The scene edge-points 2,5,7,8,10 are matched with the edge-points of the model M_{11}. The scene edge-points 1,3,6,7 are matched with the edge-points of the model M_{1K}. The scene edge-points 4,11,14 are matched with the edge-points of the model M_{nr}

Appendix

Assume that the edge-points of the model objects and scene are detected and marked by integers 1, 2, 3, ...

Consider Table 4. Each entry of Table 4 represents the Euclidean measure J_1 between the principal curvatures of an edge-point of a model object (at a particular view) and an edge-point of the scene. A threshold T_{J_1} decides whether a scene edge-point matches a model edge-point. According to the numerical value of T_{J_1} a particular scene edge-point may be matched with more than one edge-points of different views of a particular model and/or different models of different objects and vice versa. This initial matching between scene edge-point and model edge-point segregates the most dissimilar features between the scene and model objects. The symbol ⊙ of Table 4 indicates the similarity between the ith edge-point of the scene and the jth edge-point of Kth view of the rth model. After the initial segregation of dissimilar features we decompose Table 4 into n-number of tables where n represents the total number of views of the model objects. On the individual table we separately apply the minimal spanning tree algo-

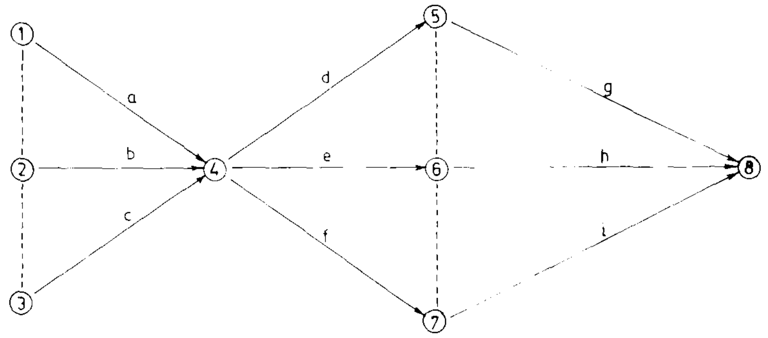


Figure 9. a = perform the task of initial edge-point matching of the scene with respect to the model M_{11}; b = perform the task of initial edge-point matching of the scene with respect to the model M_{1k}; c = perform the task of initial edge-point matching of the scene with respect to the model M_{nr}; d = determination of the compatible structure (using minimum spanning tree) between the matched scene edge-points and the matched edge-points of the model M_{11}; e = determination of the compatible structure (using minimum spanning tree) between the matched scene edge-points and the matched edge-points of model M_{1k}; f = determination of the compatible structure (using minimum spanning tree) between the matched scene edge-points and the matched edge-points of the model M_{nr}; g = computation of the cluster centers $(\alpha', \beta', \gamma')$ and T_x, T_y, T_z for the best possible hypothesis about the model M_{11}; h = computation of the cluster centers $(\alpha', \beta', \gamma')$ and T_x, T_y, T_z for the best possible hypothesis about the model M_{1k}; i = computation of the cluster centers $(\alpha', \beta', \gamma')$ and T_x, T_y, T_z for the best possible hypothesis about the model M_{nr}.

rithm discussed in subsection 5.3.1. Thus we get the most compatible match between the scene edge-points and model edge-points. A node ◎ in a table would consist on a match. An arc would exist between two nodes if the two matches complied with the four compatibility constraints mentioned in subsection 5.3.1. In Figure 9 we indicate that the task of edge-point matching can be done simultaneously (i.e., parallel computation is possible).

After determining the compatible structure on each table we compute the cluster center for each compatible structure of each table. The cluster center (see Tables 1-3) at each table is essentially formed by taking the average of all the rotational parameters α, β, γ of all compatible structures of each table. (Thus in Case study 1 of Section 6 we get four cluster centers indicated in Tables 1-3). The translation parameters t_x, t_y and t_z of all the compatible structures of each table are further modified to t_x', t_y' and t_z' due to the fixing up of all the rotational parameters α, β and γ of each table to the cluster center α', β' and γ'. After this we switch over to the task of non-edge-point matching. We apply the 3-D transformation (i.e., α', β', γ' and T_x, T_y, T_z, which are already estimated for the best possible hypothesis for the edge-points) to the non-edge-points which are surrounded by the matched edge-points. We compare the transformed coordinates of the non-edge-points of the model objects with the coordinates of the corresponding non-edge-points of the scene. The comparison is performed through the Euclidean measure J_2. A threshold T_{J_2} indicates whether the transformed coordinates of the non-edge-points of the model objects are acceptable. Finally we compare (through the Euclidean measure J_3) the mean curvature and Gaussian curvature of the transformed

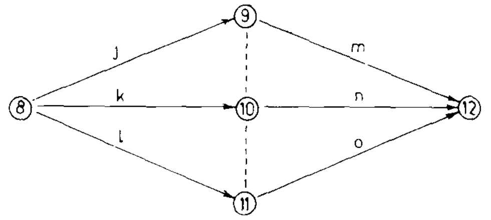


Figure 10. The end point (i.e. node 8) of Figure 9 is the starting point of Figure 10. j = application of 3-D transformation on the non-edge-points of the model M_{11}; k = application of 3-D transformation on the non-edge-points of the model M_{1k}; l = application of 3-D transformation on the non-edge-points of the model M_{nr}; m = matching (through the mean curvature and Gaussian curvature) of the non-edge-points of the model M_{11} with the non-edge-points of the scene; n = matching (through the mean curvature and Gaussian curvature) of the non-edge-points of the model M_{1k} with the non-edge-points of the scene; o = matching (through the mean curvature and Gaussian curvature) of the non-edge-points of the model M_{nr} with the non-edge-points of the scene.

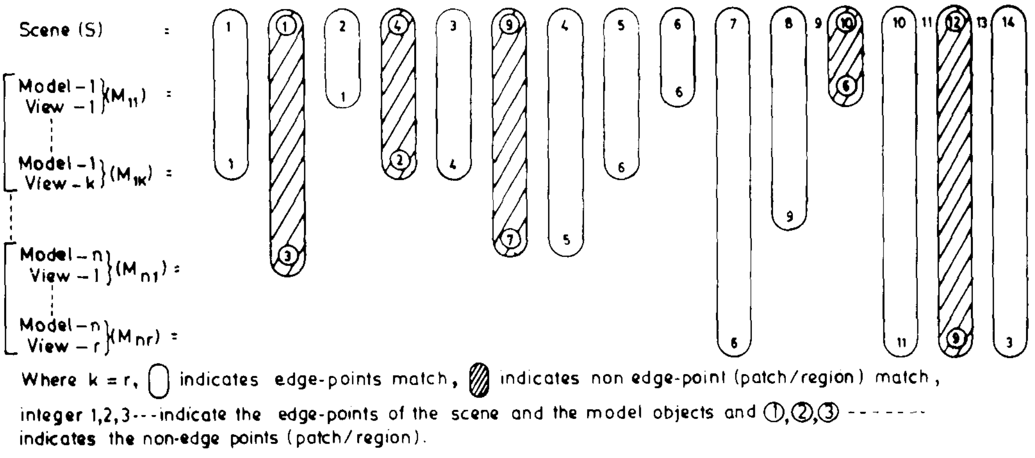

Scene (S) :
Model-1 View-1 } (M11) :
Model-1 View-k } (M1k) :
Model-n View-1 } (Mn1) :
Model-n View-r } (Mnr) :
1 ① 2 ④ 3 ⑨ 4 5 6 7 8 9 ⑩ 10 11 ⑫ 13 14
1 1 ② 4 ③ 6 ⑤ 6 ⑦ 5 9 6 11 ⑨ 3
Where k = r, ▯ indicates edge-points match, ▨ indicates non edge-point (patch/region) match,
integer 1,2,3---indicate the edge-points of the scene and the model objects and ①,②,③ -------
indicates the non-edge points (patch/region).

Figure 11.

non-edge-points of the model objects with those of the corresponding non-edge-points of the scene. A threshold T_{J_3} indicates whether a non-edge-point of the scene matches a non-edge-point of the model objects. Figure 10 indicates the possibility of parallel computation of the task of non-edge-point matching.

Thus, for an arbitrary example we get the complete compatible structure shown in Figure 11.

References

Altschuler, M.D., J.L. Posdamer, G. Frider, B.R. Altschuler and J. Taboada (1981). The numerical stereo camera. *Proc. Soc. for Photo-Optical Instrumentation Engineers Conf. on 3-D Machine Perception, SPIE* 283. SPIE, Bellingham, 15-24.

Agin, G.J. (1972). Representation and description of curved objects. Tech. Rep. AIM-173, Stanford University.

Besl, P.J. and R.C. Jain (1984). Surface characterization for three-dimensional object recognition. RSD-TR-20-84, Electrical Engineering and Computer Science Dept., University of Michigan, Ann Arbor, MI (Dec.).

Besl, P.J. and R.C. Jain (1986). Invariant surface characteristics for 3-D object recognition in range images. *Computer Vision, Graphics, and Image Processing* 33, 33-80.

Besl, P.J. and R.C. Jain (1985). Three-dimensional object recognition. *Comput. Surv.* 17(1), 75-145.

Ballard, D.H. (1983). Viewer independent shape recognition. *IEEE Trans. Pattern Anal. machine Intell.* 5(4), 653-660.

Ballard, D.H. and C.M. Brown (1982). *Computer Vision.* Prentice-Hall, Englewood Cliffs, NJ.

Bolles, R.C. and R.A. Cain (1982). Recognizing and locating partially visible objects: the focus feature method. *Int. J. Robotics Res.* 1(3), 57-81.

Bhanu, B. (1982). Surface representation and shape matching of 3-D objects. *Proc. Pattern Recognition and Image Processing Conf.* (Las Vegas, NV, June 14-17). IEEE, New York, 344-354.

Bhanu, B. (1984). Representation and shape matching of 3-D objects. *IEEE Trans. Pattern Anal. Machine Intell.* 6(3), 340-350.

Boyter, B. and J.K. Aggarwal (1984). Recognition with range and intensity data. *Proc. Workshop on Computer Vision: Representation and Control* (Annapolis, MD, Apr. 30-May 2). IEEE, New York, 112-117.

Coleman, E.N. and R. Jain (1982). Obtaining shape of textured and specular surfaces using four-source photometry. *Computer Graphics and Image Processing* 18(4), 309-328.

Carnahan, B., H.A. Luther and J.O. Wilkes (1969). *Applied Numerical Methods.* Wiley, New York.

Cline, A.K. (1981). Surfac smoothing by splines under tension. Tech. Rep. CNA-168, University of Texas at Austin, Dept. of Computer Science.

DoCarmo, M.P. (1976). *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Englewood Cliffs, NJ, 134-156.

Faugeras, O.D. (1984). New steps towards a flexible 3-D vision system for robotics. *Proc. 7th Intl. Conf. on Pattern Recognition* (Montreal, Canada, July 30-Aug. 2). IEEE, New York, 796-805.

Fischler, M. and R. Bolles (1981). Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Computer Vision, Graphics, and Image Processing* 24(6), 381-395.

Gennery, D.B. (1979). Object detection and measurement using stereo vision. *Proc. 6th Intl. Joint Conf. on Artificial Intelligence* (Tokyo, Japan, Aug. 20-23), 320-327.

Grimson, W.E.L. (1980). A computer implementation of a theory of human stereo vision. MIT Artificial Intelligence Lab Memo 565. MIT, Cambridge, MA.

Grimson, W.E.L. and T. Lozano-Pérez (1985). Recognition and localization of overlapping parts from sparse data in two and three dimensions. *Proc. IEEE Int. Conf. Robotics* (St. Louis, MO, March), 61-66.

Hsiung, C.C. (1981). *A First Course in Differential Geometry.* Wiley-Interscience, New York.

Hakalathi, H., D. Harwood and L. Davis (1984). Two-dimensional object recognition by matching local properties of contour points. *Pattern Recognition Letters* 2(4), 227-234.

Haraud, P. and R.C. Bolles (1984). 3DPO's strategy for matching three-dimensional objects in range data. *Proc. Intl. Conf. on Robotics* (Atlanta, GA, March 13-15). IEEE, New York, 78-85.

Horn, B.K.P. (1977). Understanding image intensities. *Artificial Intelligence* 8(2), 201-231.

Ikeuchi, K. and B.K.P. Horn (1981). Numerical shape from shading and occluding boundaries. *Artificial Intelligence* 17, 141-184.

Jain, R. (1983). Dynamic scene analysis. In: A. Rosenfeld and L. Kanal, Eds., *Progress in Pattern Recognition, Vol. 2.* North-Holland, Amsterdam.

Koch, M.W. and R.L. Kashyap (1985). Computer vision algorithms to recognize and locate partially occluded objects. Eng. Res. Center, Purdue University, Tech. Rep. TR-ERC-86-2, Dec.

Kanade, T. (1981). Recovery of the three-dimensional shape of an object from a single view. *Artificial Intelligence* 17, 409-460.

Kuan, D.T. and R.J. Drazaich (1984). Model-based interpretation of range imagery. *Proc. Nat. Conf. on Artificial Intelligence* (Austin, TX, Aug. 6-10). AAAI, New York, 210-215.

Nevatia, R. and T.O. Bingord (1977). Description and recognition of curved objects. *Artificial Intelligence* 8(1), 77-78.

Oshima, M. and Y. Shirai (1981). Object recognition using three-dimensional information. *Proc. 7th Intl. Joint Conf. on Artificial Intelligence* (Vancouver, B.C., Canada, Aug. 24-28), 601-606.

Oshima, M. and Y. Shirai (1983). Object recognition using three-dimensional information. *IEEE Trans. Pattern Anal. Machine Intell.* 5(4), 353-361.

Oshima, M. and Y. Shirai (1979). A scene description method using three-dimensional information. *Pattern Recognition* 11, 9-17.

Plastock, A. Roy and G. Kalley (1986). *Computer Graphics.* Schaum's Outline Series in Computers. McGraw-Hill, New York.

Ponce, J. and M. Brady (1987). Towards a surface primal sketch. In: T. Kanade, Ed., *Three-dimensional Machine Vision.* Kluwer Academic Publ., Dordrecht, 195-240.

Ray, K.S. and D. Dutta Majumder (1989). Application of differential geometry to recognize and locate partially occluded objects. *Pattern Recognition Letters* 9, 351-360.

Smith, D.R. and T. Kanade (1984). Autonomous scene description with range imagery. *Proc. Image Understanding Workshop* (New Orleans, LA, Oct. 3-4). DRAPA, McLean, VA, 282-290.

Stockman, G., S. Kopstein and S. Benelt (1982). Matching images to models for registration and object detection via clustering. *IEEE Trans. Pattern Anal. Machine Intell.* 4(3), 229-241.

Shamos, M.I. and D. Hoey (1975). Closest point problems. *16th Annual IEEE Symp. Foundations of Computer Science,* Oct., 157-162.

Ullman, S. (1979). *The Interpretation of Visual Motion.* MIT Press, Cambridge, MA.

Vemuri, B.C., A. Mitiche and J.K. Aggarwal (1987). 3-D object representation from range data using intrinsic surface properties. In: T. Kanade, Ed., *Three-Dimensional Machine Vision.* Kluwer Academic Publ., Dordrecht, 241-266.

Witkin, A.P. (1981). Recovering surface shape and orientation from texture. *Artificial Intelligence* 17, 17-45.

Woodham, R.J. (1981). Analysing images of curved surfaces. *Artificial Intelligence* 17, 117-140.