SURFACE REPRESENTATION AND SHAPE MATCHING OF 3-D OBJECTS

Bir Bhanu

Digital Systems Department
Ford Aerospace & Communications Corporation
Newport Beach, CA 92660

ABSTRACT

A 3-D scene analysis system for the shape matching of real world 3-D objects is presented. A 3 point seed algorithm is used to approximate a 3-D object by a set of planar faces. The 3-D model of an object is obtained by combining the object points from a sequence of range data images corresponding to various views of the object, applying the transformations and then approximating the surface by polygons. A hierarchical stochastic labeling technique is used to match the individual views of the object taken from any vantage point. The results of partial shape recognition are used to determine the orientation of the object in 3-D space. Examples are given using several unknown views of a complicated automobile casting used in the suspension system.

I. INTRODUCTION

The direct measurement of range simplifies many of the problems of 2-D scene analysis. It is very useful in 3-D shape analysis and segmentation. In this paper, the representation and modelling aspects of 3-D scene analysis will be considered. A method based on a laser triangulation to acquire 3-D data will be described. The problems related with 3-D data acquisition and geometric processing will be addressed. A technique for representing a 3-D object by a set of planar convex faces will be presented. This technique is used to generate a 3-D model of an object. A hierarchical stochastic labeling is used to do shape matching of 3-D objects. The matching is called "face matching" i.e., a partial 3-D shape is recognized as an approximate match to a part of a larger 3-D shape. Results are presented on a complex automobile part.

II. 3-D SCENE ANALYSIS SYSTEM

Fig. 1 shows the schematic diagram of the 3-D scene analysis system implemented in this work. The object surface point data was obtained with a laser ranging system whose principle is shown in Fig. 2. A laser emits a beam of ruby red light which is reflected by a mirror which rotates and sweeps the beam along the x-axis to produce one scan line. The beam is reflected from the object, and the z-distance is calculated from the location of the maximum response in each bank of detectors. The platform on which the object rests can be raised or lowered (this is the y-axis) and can also be rotated (around the y-axis). The sampling distances used here are 3.0 mm in the x-axis, 2.0 mm in the y-axis, and an accuracy of 0.5 mm in the z-distance is achieved.

The data so obtained is in the observer centered coordinate system. While creating a 3-D model of the object, object centered representation is required. This is computed by marking the zero position for x- and y-axis and obtaining a reference value for z-axis on the platform (fig. 2). Thus all the points can be transformed to the same coordinate system. In order to create a 3-D model of the object, a range data image was produced for every 30° rotation of the object around the y-axis. Finally, top and bottom views of the object were taken. These two views were put in correspondence with the other views by using several control points on the object to compute the transformation.

III. REPRESENTATION AND MODELLING

"Representation" is defined as the act of making a description of an object. The description should capture an intuitive notion of shape and it should be compact and simple. The simplest approach to analyze 3-D objects is to model them as polyhedra. In order to handle curved and more complex objects, other representations and models have been investigated [4]. Binford [3] proposed the concept of a generalized cylinder (or cone) to represent curved 3-D objects. There are an infinite number of possible generalized cones representing a single object. More constraints are needed to get a unique description. Although generalized cones or volume representations imply some surface description, they fail to describe the

82CH1761-6/82/0000/0349$00.75

junctions or surface peculiarities [2]. Also one detects surfaces first from partial views, and only after several different views of the object we have enough data to obtain volume properties. Hence the need to find a suitable surface representation. It is possible to represent arbitrary shapes with generalised cones by making them arbitrarily complex, but their computation is difficult. The generalized cone primitives used in [12] are not sufficient to represent the complicated casting, as has been used in this work. Badler and Bajcsy [4] present a good discussion of the relative merits of surface and volume representations.

Our approach to the analysis of a 3-D range data image is to first extract the relevant 3-D object as sets of 3-D points and then work directly on these sets without regard to the original image. This approach frees one from a particular image. The goal is to produce a complete description (3-D model) of the surface of a 3-D object in terms of faces. Such a representation should be complete, that is, it should sample the entire surface of the object, and allow for matching of individual views of the object taken from any vantage point. An object is thus defined by a finite number of selected points in three-space. However, only the geometrical position of each point is known; no topological information is available.

Representation and models are very intimately connected. 2-D models have been commonly used in the analysis with constraints on the configuration of objects whenever possible. A tradeoff is involved between representation and modelling. 2-D models make the representation easier at the expense of a complex modelling task. 3-D models are more general, but the representation must take care of mapping it to the view-domain. They are very powerful for 3-D shape analysis. Hierarchical models which involve both 2-D and 3-D models have been used [1]. In this work we generate a 3-D model of the object in terms of planar faces approximated by polygons. The control structure used in shape matching is hierarchical and described in the next section.

<u>Surface Approximation by Polygons</u>

Representing a 3-D object as a set of planar faces approximated by polygons is a two step process (fig. 3). In the first step we find the set of points that belong to various faces of the object using a three point seed algorithm [7,8] and in the second step, approximate the face points obtained in step 1 by polygons. The three point seed method for the extraction of planar faces from range data is a model fitting method. It can be viewed as a special case of the Random Sample Consensus (RANSAC) paradigm [6].

<u>The 3-point seed method</u> : In a well-sampled 3-D object, any three points lying within the sampling distance of each other (called a 3-point seed) form a plane (called the seed plane) which: (a) coincides with that of the object face containing the points, or (b) cuts any object face containing any of the 3 points.

A seed plane satisfying (a) results in a plane from which a face should be extracted, while a seed plane satisfying (b) should be rejected. Two simple conditions that suffice to determine if a plane falls into category (b) are: convexity and narrowness.

The algorithm involves the following steps. Details about the method can be found in [7,8].

1. From the list of surface points select 3 points which are noncolinear and near relative to sampling distances.
2. Obtain the equation of the plane passing through the three points chosen in step 1.
3. Find the set of points P which are very close to this plane.
4. Apply the convexity condition to the set P to obtain a reduced convex set P'. This separates faces lying in the same plane.
5. Check the set P' obtained in step 4 for narrowness. This test eliminates small narrow planes.
6. If the face is obtained correctly (i.e., convexity and narrowness conditions are satisfied), remove the set of points belonging to this face from the list of points and proceed to step 1 with the reduced number of points in the list.

After the surface points belonging to a face have been obtained, two checks are made. (1)All the points which have been previously associated with various faces are checked for the possible inclusion in the present face. (2)The set of points of the present face is checked for the possible inclusion in the previous faces. The application of these two tests provides the points which belong to more than one face. This information in turn provides the knowledge about the neighbors of a face and relations among them. Also The method is applied in stages; the largest faces (in terms of the number of points in the face) are found first, then smaller faces on down to some minimum size. The method requires four thresholds which are tied to the sampling distances. The complexity of the 3-point seed algorithm is $O(n^2 \mathrm{Log}\ n)$ when we use k-d tree [9] in organising the data. The polygonal approximation of a face involves the following steps. (1)Get the points belonging to a face. (2)Obtain the binary image of the face points. (3)Trace the boundary of the image obtained in step 2 using a boundary follower. Obtain (x,y,z) coordinates of the boundary points of the face. (4)Perform a polygonal approximation of the boundary of the face by detecting the points of high curvature. Surface Approximation Results : Fig. 4 shows a complicated casting of an automobile piece. This object does not contain any major horizontal or vertical surface. The 14 views obtained with the range data acquisition system are shown as gray scale images in fig. 5. In this figure lighter points are farther away from the observer and the darker ones are closer. After thresholding the background points, each individual view shown in figs. 5 had approximately 2000 points. Surface points for the composite object were 8314.

The 3-point seed method was applied to the 14 individual views shown in fig. 5 and to the composite object. Fig. 6 shows the faces found for 0° view. In this figure various faces are shown in different colors. The rejected points and the points common to two or more faces (edge points) are shown in brown and white color respectively. They are labeled in the order they are found using the 3-point seed algorithm. The points that could not make up a face having at least 20 points were rejected. A rejected point lies inside some of the faces because it has been missed in the process of data acquisition. Also some of the rows have been shifted because of the continuous nature of the data. Table 1 gives the properties of faces in the 0° view. Note that a face may have no neighbors, because a face that could not possess more than a certain minimum number of points was rejected. In the model 85 faces were found. The number of faces found, and their distribution fits well with the results from the individual views.

IV. SHAPE MATCHING OF 3-D OBJECTS

Fourier descriptors and moments have been used for the recognition of 3-D shapes [7,10]. However, they are global features and are not suitable to solve the important class of problems which require the partial recognition of the shape. Milgram and Bjorklund [8] mention preliminary efforts of 3-D matching by pairing the list of planes of the reference model and the sensed image using a guided search procedure.

In the previous section we presented a discussion and our approach to representation and modelling of 3-D objects. In 3-D scene analysis work bottom up, top down and a mixture of these two have been used. The hierarchical control structure is a popular choice since it eliminates unnecessary search during the recognition process. Our approach for 3-D shape matching uses planar faces as primitives. The control structure of the 3-D shape matching algorithm used here is hierarchical in the sense that at higher levels of hierarchy more contextual information is used to accomplish the partial shape matching task.

Shape Matching Algorithm : Fig. 7 shows the block diagram of the two stage hierarchical stochastic labeling technique. Shape matching is performed by matching the face description of an unknown view with the stored model using the available contextual information. The same set of descriptors is used for the description of both the faces of the model and an unknown view. Input to the stochastic labeling process includes, features of the faces, information about the neighbors of a face, and equation of the plane describing a face.

Let $T = (T_1, T_2, \dots, T_N)$ and $O = (O_1, O_2, \dots, O_{L-1})$ be the face representation of an unknown view and the model respectively, where T_i and O_j are planar faces, $i = 1,\dots,N$ and $j = 1,\dots,L-1$. The elements of the unknown view will be referred to as units and elements of the model as classes. We want to identify an unknown view within the model. We are therefore, trying to label each of the faces of an unknown view T_i $(i = 1,\dots,N)$ eitner as a face O_j $(j = 1,\dots,L-1)$ or as not belonging to the model O (label O_L = Nil). Each face T_i of the unknown view therefore has L possible labels.

To each of the units T_i, we assign a probability $p_i(\ell)$, $\ell = 1,\dots,L$ that the unit belongs to class O_k. This is conveniently represented as a probability vector $\vec{p}_i = [p_i(1),\dots,p_i(L)]^T$. The set of all vectors $\vec{p}_i$ $(i = 1,\dots,N)$ is called a stochastic labeling of the set of units. Units are related to one another through their neighbors. The set of units related to T_i is denoted by V_i. In order to compare the local structure of T and O, the world model is specified by the compatibility functions C_1 and C_2, which are defined over a subset $S_1 \subseteq (N \times L)^{(2)}$ and $S_2 \subseteq (N \times L)^{(3)}$ for the first and second stage of the hierarchy respectively. At the first stage $C_1(T_i, O_k, T_j, O_\ell)$ measures the adequacy of calling unit T_i as O_k and unit T_j as O_ℓ where $T_j \in V_i$. Similarly at the second stage $C_2(T_i, O_k, T_{i_1}, O_{\ell_1}, T_{i_2}, O_{\ell_2})$ measures the adequacy of calling unit T_i as O_k, unit T_{i_1} as O_{ℓ_1} and T_{i_2} as O_{ℓ_2} where T_{i_1} and $T_{i_2} \in V_i$. We also define a compatibility vector $\vec{q}_i = [q_i(1),\dots,q_i(L)]^T$ for all the units at each of the labels of hierarchy. Intuitively this tells us what $\vec{p}_i$ should be given $\vec{p}_j$ at the related units and the compatibility function. For simplicity we shall denote $C_1(T_i, O_k, T_j, O_\ell)$ as $C_1(i,k,j,\ell)$ and $C_2(T_i, O_k, T_{i_1}, O_{\ell_1}, T_{i_2}, O_{\ell_2})$ as $C_2(i,k,i_1,\ell_1,i_2,\ell_2)$. Mathematically,

$$q_i^{(j)}(k) = \frac{Q_i^{(j)}(k)}{\sum_{\ell=1}^{L} Q_i^{(j)}(\ell)}, \quad j = 1,2 \quad k = 1,\ldots,L \tag{1}$$

where, at the first stage,

$$Q_i^{(1)}(k) = \sum_{j \in V_i} \sum_{\ell=1}^{L} C_1(i,k,j,\ell) p_j(\ell) \quad i = 1,\ldots,N \quad k = 1,\ldots,L \tag{2}$$

and at the second stage,

$$Q_i^{(2)}(k) = \sum_{\ell_1,\ell_2=1}^{L} C_2(i,k,i_1,\ell_1,i_2,\ell_2) p_{i_1}(\ell_1) p_{i_2}(\ell_2) \tag{3}$$

$$i = 1,\ldots,N; \; k = 1,\ldots,L \text{ and } i_1,i_2 \in V_i$$

As discussed in [7,11] the global criteria that measure the consistency and ambiguity of the labeling over the set of units are given by,

$$J^{(j)} = \sum_{i=1}^{N} \vec{p}_i \cdot \vec{q}_i^{(j)}, \quad j = 1,2 \tag{4}$$

The maximization of the criteria is done using the gradient projection method. Using the initial probability assignment $\vec{p}_i(0)$ $(i = 1,\ldots,N)$ first we maximize $J^{(1)}$ and then use these results to maximize $J^{(2)}$ [7,11].

Initial Assignment of Probabilities

The initial probabilities are computed using the following features of a face. Area, perimeter, length of the maximum, minimum and average radius vectors from the centroid of a face, number of vertices in the polygonal approximation of the boundary of a face, angle between the maximum and minimum radius vectors, and ratio of Area/Perimeter^2 of a face. Let P be the number of features used. We measure the quality of correspondence between the faces T_i and O_k as

$$M(T_i,O_k) = \sum_{p=1}^{P} |f_{tp} - f_{op}| W_p \tag{5}$$

where,

f_{tp} = pth feature value for the face of an unknown view

f_{op} = pth feature value for the face of the model

W_p = weight factor for the pth feature

The initial probabilities are chosen proportional to $1/(1+M(T_i,O_k))$ and normalized so that they sum to 1.

Computation of Compatibilities : The compatibility of a face of an unknown view with a face in the model is obtained by finding transformations, applying them and computing the error in feature values [7]. These transformations are based upon,

1) Scale, the ratio of area of two faces.
2) Translation, difference in the centroidal coordinates of the two faces.
3) Orientation, difference in the orientation of two faces so that they are in the same plane.
4) Rotation, to obtain maximum area of intercept, once the two faces are in the same plane.

The problem of defining $p_i(nil)$, C_1 and C_2 when some of the faces in the unknown view are matched to the nil class is solved in [7].

Examples : The evaluation of the compatibility vector $q_i^{(j)}$ $(j = 1,2)$ requires the knowledge about the neighbors of a face (see eq. 1). Normally we have considered the number of neighbors to be 1 or 2 in the computation of $q_i^{(1)}$ and 2 largest neighbors in the computation of $q_i^{(2)}$ for all the units. If a unit has only one neighbor while computing $q_i^{(2)}$, then compatibility C_1 is used instead of C_2 in its computation. If a unit has no neighbors, then this unit is firmly assigned to the best matched class at the time of computation of initial probabilities. In matching only the best 29 faces of the model are considered in order to reduce the complexity of the matching task. In testing the shape matching algorithm we consider 3 unknown view figs. 5(a), 5(b) and 5(1) corresponding to 0°, 30°, and 330° respectively. Although the model is obtained using these views, the model does not contain all the faces corresponding to any unknown view. This is because of the nature of how the surface points corresponding to the composite object were obtained. Therefore, the use of these unknown views is justified for the evaluation of the shape matching technique.

Example 1 : Fig. 6 shows the faces found in the view shown in fig. 5(a). Table 1 shows the neighbors of the faces. The neighbors are arranged by size in descending order. Table 2 shows the results of labeling at different iterations. One way of checking the results of labeling is to compute the relative orientation of the object in 3-space using the final assignment of units. Several triple of units such as (1, 2, 3), (1,3,4), (1, 13, 16), (16, 17, 18), (5, 6, 7), (2, 3 ,4), (2, 17, 18) etc. produce the transformation matrix T of direction cosines very accurately. For example, the triple (5, 6, 7) solves T as

$$\begin{bmatrix} 1.0000 & -0.00305 & -0.011019 \\ -0.00534 & 1.00000 & 0.00586 \\ -0.009175 & 0.00000 & 1.00000 \end{bmatrix}$$

The arccos of each coefficients of the matrix T leads to,

$$\begin{bmatrix} 0 & 90.14 & 90.63 \\ 90.30 & 0 & 89.66 \\ 90.52 & 90.00 & 0 \end{bmatrix}$$

Thus the relative orientation of the unknown view with respect to the model is about 0 degrees. Ideally the matrix T should be identity for this example.

Example 2 : Fig. 8 shows the faces found in the view shown in Fig. 5(b). There are 24 faces in this view and they are labeled in the order thay are found. Comparing figs. 6 and 8, it can be seen how some of the faces of fig. 8 should be labeled. For example faces 11 and 7 in fig. 8 correspond to faces 8 and 10 in fig. 6 respectively. Similarly the correspondence for other faces can be obtained and the matching results shown in Table 2 can be verified. Using the triple (4,7,8), matrix T is obtained as

$$\begin{bmatrix} 0.88383 & 0.09058 & 0.46854 \\ -0.20947 & 1.00000 & -0.19653 \\ -0.45183 & 0.01863 & 0.86441 \end{bmatrix}$$

For a relative orientation of 30 degrees in the x-z plane the matrix T is,

$$\begin{bmatrix} 0.866 & 0 & 0.500 \\ 0 & 1 & 0 \\ -0.500 & 0 & 0.866 \end{bmatrix}$$

Thus the relative rotation in the x-z plane of about 30 degrees is obtained.

Example 3 : Fig. 9 shows the faces obtained in the view shown in fig. 5(1). There are 24 faces in this view and as before they are labeled in the order they are found. Comparing figs. 6, 8 and 9 one can observe how the face description has changed. Also it can be inferred how the faces of fig. 9 should be labeled with respect to the labeling of the faces in figs. 6 and 8. Table 2 shows the results. Most of the labels are correct, but a few of them are wrong because of the higher degree of similarity of the local structure of the incorrect match. As in the previous examples, triples of units can be used to compute the matrix T [7].

V. CONCLUSIONS

The results of labeling are good and very reasonable. A few incorrect assignments result because the structure and description of a unit with its neighbors matches better with incorrect match than the correct match. Also if the object has some symmetry, it is likely that there will be multiple matches. The results of labeling depend upon the planar surface approximation and neighborhood information. The number of views to obtain a model depends upon the complexity of the object. Normally, we used 3 iterations at the first stage and 4 to 8 iterations at the second stage. For the examples presented the computation time varied from about 4 minutes to 20 minutes on a PDP-10. Over 95% of this time is spent in the computation of rotation needed in the compatibility computation. This is because we store only the boundary of the image of a face. Also we do not store the compatibility values, and recompute them when the gradient is required. If we store the images of the faces and the compatibility values, computation time will be much smaller. These matching results could be useful in controlling a robot manipulator on an assembly line or inspection stages of the production.

REFERENCES

[1] Y. Shirai, "Recent Advances in 3-D Scene Analysis," Proc. 4th IJCPR Kyoto, Japan, Nov. 1978, pp. 86-94.

[2] R. Bajcsy, "Three Dimensional Scene Analysis," Proc. 5th ICPR, Miami Beach, Florida, Dec. 1980, pp. 1064-1074.

[3] T.O. Binford, "Visual Perception by Computer," IEEE Conf. on Sys. and Control, Miami, Dec. 1971.

[4] N. Badler and R. Bajcsy, "Three-Dimensional Representations for Computer Graphics and Computer Vision," ACM Comp. Graphics, Vol. 12, pp. 153-160, Aug. 1978.

[5] D.L. Milgram and C.M. Bjorklund, "Range Image Processing : Planar Surface Extraction," Proc. 5th ICPR, Miami Beach, Florida, Dec. 1980, pp. 912-919.

[6] M.A. Fischler and R.C. Bolles, "Random Consensus: A Paradigm for Model Fitting With Applications in Image Analysis and Automated Cartography," Comm. of ACM, Vol. 24, pp. 381-395, June 1981.

[7] B. Bhanu, "Shape Matching and Image Segmentation Using Stochastic Labeling," Ph.D. Thesis, USCIPI Report 1030, Image Processing Institute, USC, Los Angeles, Ca. August 1981.

[8] T.C. Henderson and B. Bhanu, "Three Point Seed Method for the Extraction of Planar Faces From Range Data," Proc. IEEE Int. Conf. on Industrial Applications of Machine Vision, Raleigh, N.C., May 3-5, 1982.

[9] J.H. Friedman et.al., "An Algorithm for Finding Best Matches in Logarithmic Expected Time," ACM Trans. on Math. Software, Vol. 3, pp.209-226, Sept. 1977.

[10] T.P. Wallace, O.R. Mitchell and K. Fukunaga, "Three-Dimensional Shape Analysis Using Local Shape Descriptors," IEEE Trans. on PAMI Vol. PAMI-3, pp. 310-323, May 1981.

[11] B. Bhanu and O.D. Faugeras, "Shape Matching of 2-D objects using a Hierarchical Stochastic Labeling Technique," Proc. PRIP 82 , Las Vegas, Nevada, June 15-17, 1982.

[12] R. Nevatia and T.O. Binford, "Description and Recognition of Curved objects," Artificial Intelligence, Vol. 8, pp. 77-98.

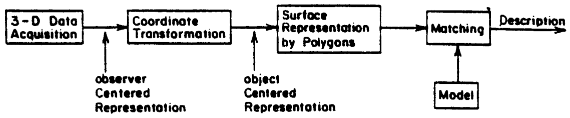

Fig. 1. The Schematic diagram of 3-D Scene Analysis System.

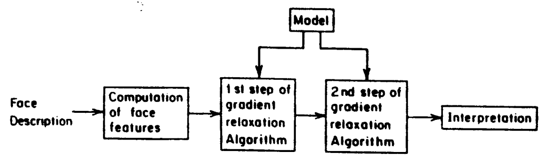

Fig. 7. Block diagram of the 3-D Shape Matching Algorithm.

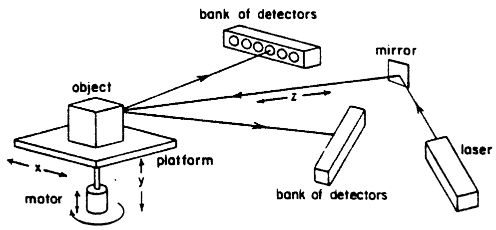

Fig. 2. Laser Ranging System.

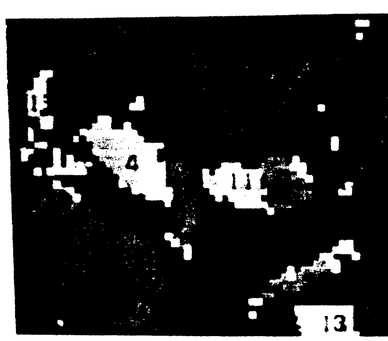

Fig. 8. Faces found in the view shown in Fig. 5(b). There are 24 faces in this view and they are labeled in the order they are found.

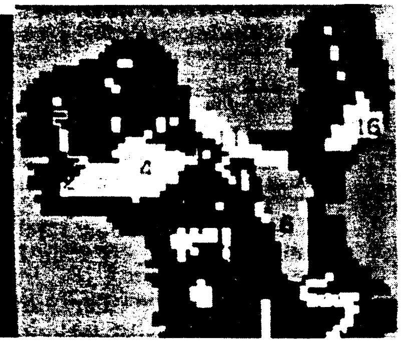

Fig. 9. Faces found in the view shown in Fig. 5(1). There are 24 faces in this view and they are labeled in the order they are found.

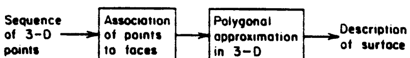

Fig. 3. Two step process to approximate surface by polygons.

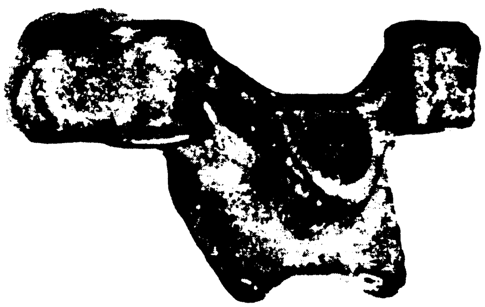

Fig. 4. Automobile piece analyzed.

Fig. 5. The 14 range data views of the object shown as gray scale images. The lighter points are away from the observer and the darker ones are closer. They are labeled from a to n.

Fig. 6. Faces found in the view shown in Fig. 5(a). There are 22 faces in this view and they are labeled in the order they are found. The rejected points and the points common to two or more faces are shown in brown and white color respectively.

Table 1. Faces and Their Neighbors in the 0^{o} View

FACE	NO. OF POINTS	NEIGHBORS		
1	103	12	17	0
2	84	3	13	18
3	176	2	9	0
4	93	5	0	0
5	94	4	9	0
6	84	15	0	0
7	105	10	8	13
8	64	7	10	0
9	70	3	5	0
10	83	7	8	21
11	67	16	21	0
12	52	1	17	0
13	60	7	2	18
14	43	19	0	0
15	50	6	20	0
16	49	11	22	21
17	35	1	12	0
18	30	2	13	0
19	23	14	0	0
20	36	15	0	0
21	24	10	11	16
22	31	16	0	0

Table 2. Labels of the Faces

Face Number	Labels at Different Iterations											
	Ex. 1, Fig. 6				Ex. 2, Fig. 8				Ex. 3, Fig. 9			
	0	3	1	6	0	3	1	7	0	3	1	8
1	86	1	1	1	86	86	86	1	86	5	86	86
2	86	2	2	2	86	86	2	2	2	2	2	2
3	3	3	3	3	86	2	86	2	86	86	86	3
4	86	86	86	4	86	5	5	5	86	5	5	5
5	86	5	5	5	86	86	86	5	86	6	6	6
6	86	6	6	6	86	86	86	86	86	7	7	7
7	7	7	7	7	86	86	86	24	86	86	7	7
8	21	21	21	21	86	20	20	20	86	86	5	5
9	86	22	22	22	86	34	34	34	86	86	86	86
10	24	24	24	24	86	35	35	45	86	86	24	21
11	25	25	25	25	86	34	34	53	86	86	86	33
12	86	86	86	86	86	26	26	53	86	86	86	86
13	86	33	33	33	86	86	40	53	86	86	86	27
14	86	86	86	86	86	86	64	86	86	86	86	34
15	86	86	86	86	86	86	86	86	86	86	86	46
16	86	53	53	53	86	86	41	86	86	34	86	34
17	86	86	45	45	86	86	86	62	86	46	46	46
18	46	46	46	46	86	77	86	86	86	86	79	79
19	86	86	86	50	86	86	65	65	86	86	86	70
20	86	86	86	34	86	86	86	52	86	86	86	86
21	86	56	50	50	86	70	86	50	86	86	86	72
22	86	86	86	79	86	77	77	77	86	77	77	77
23					86	86	52	52	86	86	67	86
24					86	86	86	86	86	86	86	36
Value of Criteria	--	8.0	9.4	20.8	--	3.6	4.3	23.0	--	2.7	3.8	23.5
		$J^{(1)}$		$J^{(2)}$		$J^{(1)}$		$J^{(2)}$		$J^{(1)}$		$J^{(2)}$

Bir Bhanu
June 13, 1982

proceedings

PR IP 82

IEEE Computer Society
Conference on

pattern recognition and image processing

82CH1761-6
82-80366

C PUTER
S ETY
PR SS

IEEE COMPUTER SOCIETY

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

Bir Bhanu

proceedings

PRIP 82

JUNE 14–17, 1982
LAS VEGAS, NEVADA

IEEE Computer Society
Conference on

pattern recognition and image processing

82CH1761-6
82-80366

Additional copies may be ordered from
IEEE Order Dept.
445 Hoes Lane
Piscataway, NJ 08854

IEEE Computer Society
10662 Los Vaqueros Circle
Los Alamitos, CA 90720

IEEE COMPUTER SOCIETY PRESS

IEEE COMPUTER SOCIETY

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.