Pattern Recognition Letters 11 (1990) 215-223
North-Holland

March 1990

# Segmentation of three-dimensional surfaces

Jayanta MUKHERJEE, B.N. CHATTERJI
*Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur-721302, India*

P.P. DAS
*Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur-721302, India*

Received 2 January 1989
Revised 22 June 1989

*Abstract:* In 3-D scene analysis usually the 3-D surfaces are segmented into planar and quadratic surface regions. In this paper the concept of a digital neighborhood plane has been introduced for the segmentation of digital 3-D surfaces into planar regions. The algorithm is simple and is illustrated with the help of several examples.

*Key words:* Segmentation, region growing, scene analysis, Digital Neighborhood Plane, 3-D images.

## 1. Introduction

In recent years, digital geometry has gained considerable importance due to its applications in the area of pattern recognition and image processing. The concepts of digital geometry have been widely used for object reconstruction, for building up object models (for example, see (Bhanu, 1984); (Vemuri and Aggarwal, 1984); (Dane and Bajcsy, 1982); (Boissonnat and Faugeras, 1981, 1982); (Faugeras, 1984); (Faugeras and Hebert, 1983)) and for the analysis of the range image ((Henderson, 1982); (Inokuchi et al., 1982), etc.). In the analysis, recognition and reconstruction of 3-D surfaces one tries to segment the 3-D surface into planar and quadratic surfaces (Shapiro and Freeman, 1977). Milgrim and Bjorklund (1980) have described the planar surface extraction from a range image denoted in a spherical co-ordinate system. They represented the image in a cartesian coordinate system. For each pixel, a plane is fitted into its surrounding $5 \times 5$ window and the position variable, two normal vector orientation angles and the planar fit error are computed. These data are used to form the connected components of pixels satisfying the planarity constraint. Hebert and Ponce (1982) proposed a method of segmenting depth maps into planar, cylindrical and conical primitives. Inokuchi et al. (1980) found out a ring operator for an edge region segmentation of depth maps. Henderson and Bhanu (1982) proposed a three point seed method for the extraction of planar surfaces from the range data. They gave a kind of region growing technique where the region started with a seed-plane consisting of 3 neighboring points. Then keeping a check over the convexity and narrowness of the growing region from the seedplane, a convex planar surface can be extracted.

In this paper, a straightforward concept of digital neighborhood plane for the point $p$ in 3-D space will be described. The utility of the concept for the segmentation of 3-D surfaces will be given.

## 2. Digital neighbourhood planes

Before dealing with the theory and the algorithm of segmentation, we first need to make an introduction to the core concept of this paper, that is, the Digital Neighbourhood Plane (DNP). So in this section we start by defining the DNP and a few related quantities and the properties that hold over them. Throughout the ensuing discussion we denote single points by lower case letters and sets of points by upper case letters.

Let $Z^3 = \{x = (x_1, x_2, x_3)$: $x_1$, $x_2$, $x_3$ are integers$\}$ be the 3-D digital space. A point $x \in Z^3$ can have 26 different neighbours, as shown in Figure 1, in its $3 \times 3 \times 3$ neighbourhood

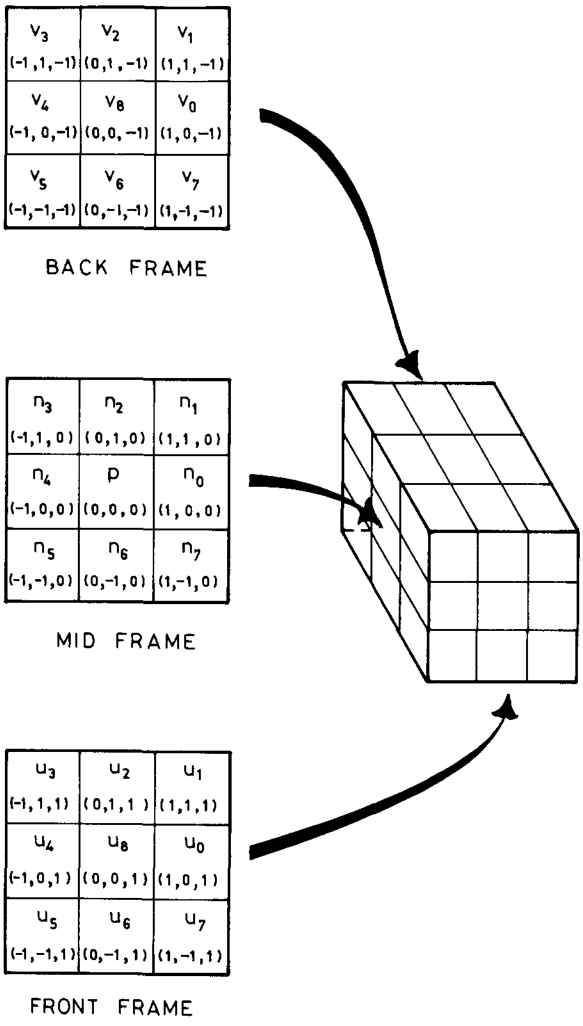


Figure 1. Neighborhood of a point $p$.

$$N(x) = \{y: |x_i - y_i| \leqslant 1, 1 \leqslant i \leqslant 3\}.$$

The neighbours are however classified into three categories, 6, 18, 26 neighbours, according to their relative coordinates as follows:

A point $y \in Z^3$ is a 6 (18,26) neighbour of $x$ if $x$ and $y$ differ in at most one (two, three) coordinates by unity.

If $x, y \in Z^3$ and $y \in N(x)$, then we also write it as a binary relation $xNy$. Clearly the relation $N$ is symmetric but it is not transitive. In an image $A \subset Z^3$, two points $x, y \in A$ are said to be $h$-connected, $h = 6$, 18 or 26, if there is a sequence of points $u_0 = x$, $u_1, u_2, \ldots, u_t = y$, such that $u_i \in A$, $0 \leqslant i \leqslant t$ and $(u_i, u_{i+1})$ are $h$-neighbours, $0 \leqslant i \leqslant t-1$. The image as a whole is said to be $h$-connected if for all $x, y \in A$, $x$ is $h$-connected to $y$. Two images $A$ and $B$ are called adjacent if there is a point $x$ in $A$ and a point $y$ in $B$ such that $xNy$.

For $xNy$ the unit vector along the straight line $xy$ in the continuous space is called the gradient from $x$ to $y$ and denoted by $\boldsymbol{xy}$. There are thus, 26 different gradients in 3-D. Using the continuity of identical gradient a (digital) straight line from $x$ to $y$ in $Z^3$ is defined as a sequence of points $u_0 = x$, $u_1, u_2, \ldots, u_t = y$, such that $u_i N u_{i+1}$, $0 \leqslant i \leqslant t-1$ and $\boldsymbol{u_{i-1}u_i} = \boldsymbol{u_i u_{i+1}} = \boldsymbol{m}$ (constant), $1 \leqslant i \leqslant t-1$.

Corresponding to 26 gradients, we count 13 distinct straight lines in 3-D by assuming $\boldsymbol{m}$ and $-\boldsymbol{m}$ to refer to the same straight line. The gradient of a straight line $L$ is $\boldsymbol{m}$ and that of straight line $Q$, perpendicular to $L$ is $\boldsymbol{n}$ where $\boldsymbol{m} \cdot \boldsymbol{n} = 0$ and '$\cdot$' denotes the dot product of vector algebra.

A plane $P$ is a connected set of points such that every straight line in $P$ with gradient $\boldsymbol{m}$ is perpendicular to a given gradient $\boldsymbol{n}$ (i.e. $\boldsymbol{m} \cdot \boldsymbol{n} = 0$), the normal gradient of $P$. For $x \in P$, any straight line of gradient $\boldsymbol{n}$ containing $x$ is called a normal to $P$ at $x$.

For a $3 \times 3 \times 3$ digital cube at $p$, there are 9 planes containing $p$. They have different orientations and their normals through $p$ are the eighteen-connected straight lines in the cube. These planes are termed as the digital neighbourhood planes (DNP's) of $p$. They are illustrated in Figure 2. The corresponding sets of points are shown in Figure 3. The neighbouring condition of $p$ will determine in which of these planes $p$ will lie.

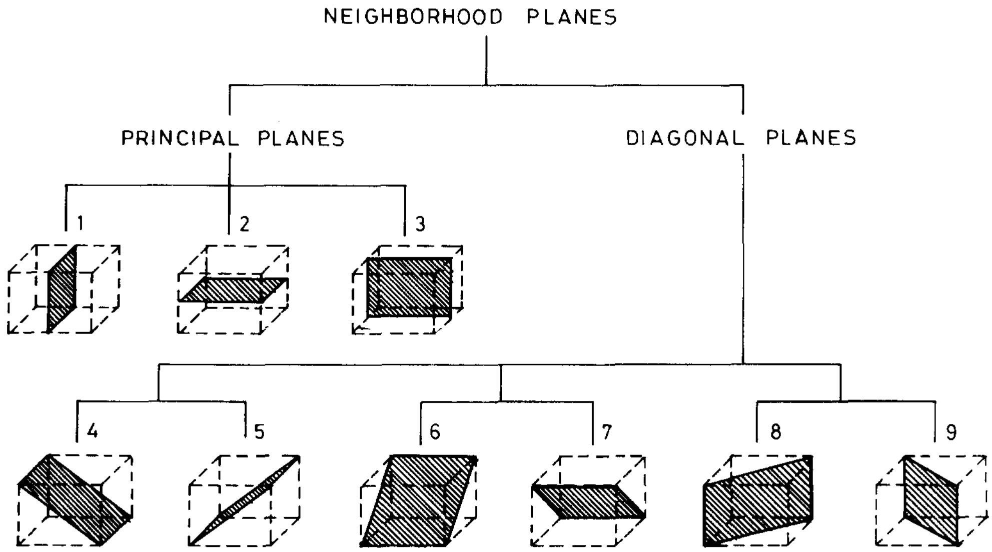


Figure 2. Neighborhood planes of $p$.

Let $P_i$ be the set of points assigned to the $i$-th neighbouring plane as described in Figure 3. Now given an image $A$ in $Z^3$, the Neighbouring Plane Set (NPS) of point $p$ (with respect to $A$) is defined as:

$$[p]_k = \{i\colon |N(p) \cap A \cap P_i| > k\}, \quad k \geqslant 3, \tag{1}$$

| | | |
|---|---|---|
| $u_2$ | $n_2$ | $v_2$ |
| $u_8$ | $p$ | $v_8$ |
| $u_6$ | $n_6$ | $v_6$ |

(1)

| | | |
|---|---|---|
| $v_4$ | $v_8$ | $v_0$ |
| $n_4$ | $p$ | $n_0$ |
| $u_4$ | $u_8$ | $u_0$ |

(2)

| | | |
|---|---|---|
| $n_3$ | $n_2$ | $n_1$ |
| $n_4$ | $p$ | $n_0$ |
| $n_5$ | $n_6$ | $n_7$ |

(3)

| | | |
|---|---|---|
| $u_3$ | $n_3$ | $v_3$ |
| $u_8$ | $p$ | $v_8$ |
| $u_7$ | $n_7$ | $v_7$ |

(4)

| | | |
|---|---|---|
| $u_1$ | $n_1$ | $v_1$ |
| $u_8$ | $p$ | $v_8$ |
| $u_5$ | $n_5$ | $v_5$ |

(5)

| | | |
|---|---|---|
| $u_3$ | $n_4$ | $v_5$ |
| $u_2$ | $p$ | $v_6$ |
| $u_1$ | $n_0$ | $v_7$ |

(6)

| | | |
|---|---|---|
| $u_5$ | $n_4$ | $v_3$ |
| $u_6$ | $p$ | $v_2$ |
| $u_7$ | $n_0$ | $v_1$ |

(7)

| | | |
|---|---|---|
| $u_1$ | $n_2$ | $v_3$ |
| $u_0$ | $p$ | $v_4$ |
| $u_7$ | $n_6$ | $v_5$ |

(8)

| | | |
|---|---|---|
| $u_3$ | $n_2$ | $v_1$ |
| $u_4$ | $p$ | $v_0$ |
| $u_5$ | $n_6$ | $v_7$ |

(9)

Figure 3. Set of points assigned to the corresponding neighborhood planes.

where $|\cdot|$ denotes the cardinality of a finite set. The value of $k$ usually lies between 3 and 5. We will denote $[p]_k$ by $[p]$ only since $k$ will remain constant. So $[p]$ actually lists the serial numbers of those neighbourhood planes which have a sufficient (thresholded by $k$) number of points around $p$ in the image, to be considered a candidate for a local neighbouring plane feature.

The neighborhood plane set $[M]$ for a finite set of points $M$ (with respect to $A$) is defined as: if $M = \{m_1, m_2, \ldots, m_n\}$, then

$$[M] = [m_1] \cap [m_2] \cap \cdots \cap [m_n]. \tag{2}$$

The following two properties of the NPS are immediate and are used later in the segmentation.

**Lemma 1.** *If $M = P \cup Q$, then $[M] = [P] \cap [Q]$.*

**Proof.** Let $P = \{p_1, p_2, \ldots, p_l\}$ and $Q = \{q_1, q_2, \ldots, q_n\}$. Now, $M = \{m\colon (m \in P) \text{ or } (m \in Q)\}$. Then

$$[P] \cap [Q] = [p_1] \cap [p_2] \cap \cdots \cap [p_l] \cap [q_1] \cap [q_2] \cap \cdots \cap [q_n]. \tag{3}$$

Now since, $[x] \cap [x] = [x]$,

$$[P] \cap [Q] = \bigcap_{\forall m \in M} [m] = [M]. \qquad \square$$

**Lemma 2.** *If $Q \subset P$, then $[P] \subseteq [Q]$.*

**Proof.** Since $Q \subset P$, we can write $P = Q \cup (P - Q)$. Therefore from Lemma 1, $[P] = [Q] \cap [(P-Q)]$. Hence the result. □

## 3. Framework for segmentation

Though theoretically any set $T$ of points of $Z^3$ can be 'segmented' into a disjoint collection of subsets, for our purpose of applications we assume the set under consideration will satisfy the following properties to allow meaningful segmentation.

**Property 1.** *T is connected.*

**Property 2.** *$\forall x \in T$, we have $[x] \neq \emptyset$, the null set.*

Such a set is commonly called a *structure*. The segmentation of a structure is necessarily a (minimal) partitioning of it. Hence if a structure $T$ is segmented into subsets (segments) $S_1, S_2, \ldots, S_m$ then the following properties are essential.

*Disjoint (non-overlapping) subsets*:

$$\forall i, j,\ 1 \leqslant i < j \leqslant m, \quad S_i \cap S_j = \emptyset.$$

*Completeness*:

$$\bigcup_{i=1}^{m} S_i = T.$$

Thus it defines a partition of the structure $T$. In addition, we desire that the number of segments should be as small as permissible by the feature extraction scheme to follow. Every such segment is expected to denote meaningful features. Producing a large number of segments may adversely affect the subsequent processing.

In terms of the NPS we define a segment $S$ to be a connected set of points such that (1) $[S] \neq \emptyset$, and (2) if $x \in S$ and $z \in N(x)$, then $[z] \cap [S] = [S]$ implies $z \in S$. The second property actually gives an algorithmic method to add more points to a segment from a given point $x$ in it.

We now establish a property of the segments which will be used later in guaranteeing disjoint segments.

**Lemma 3.** *If $S_1$ and $S_2$ are two segments of $T$, then $[S_2] \subseteq [S_1]$ implies that either $(S_1 \subseteq S_2)$ or $(S_1 \cap S_2 = \emptyset)$.*

**Proof.** Let $S_1 \cap S_2 \neq \emptyset$. Now $\forall x \in S_1$,

$$([x] \cap [S_1] = [S_1]) \Rightarrow ([x] \cap [S_1] \cap [S_2] = [S_1] \cap [S_2]).$$
$$\Rightarrow ([x] \cap [S_2] = [S_2]). \tag{4}$$

As $S_1 \cap S_2 \neq \emptyset$ and both of the $S_1$ and $S_2$ are connected, there exists a point $y$ in $(S_1 \cap S_2)$ and $z$ in $(S_1 - S_1 \cap S_2)$ such that $yNz$ holds, i.e., $y$ is a neighbour of $z$. But from (4) $[z] \cap [S_2] = [S_2]$. Hence, from the definition of a segment $z \in S$, which in turn implies that $z \in (S_1 \cap S_2)$, this is contradictory unless $((S_1 - S_1 \cap S_2) = \emptyset)$ or $(S_1 \cap S_2 = \emptyset)$. That is, $(S_1 \subseteq S_2)$ or $(S_1 \cap S_2 = \emptyset)$. □

A particular case follows from the above lemma.

**Corollary.** $([S_2] = [S_1]) \Rightarrow (S_2 = S_1)$ *or* $(S_1 \cap S_2 = \emptyset)$.

**Proof.** From Lemma 3, $([S_2] = [S_1]) \Rightarrow (S_2 \subseteq S_1)$ or $(S_1 \subseteq S_2)$ or $(S_1 \cap S_2 = \emptyset)$. Hence the result. □

In addition, if $S_1$ and $S_2$ are adjacent, then $[S_1] \subseteq [S_2]$ would by the definition of a segment imply that $S_1$ subsumes $S_2$. Hence the lemma.

**Lemma 4.** *If $S_1$ and $S_2$ are adjacent, then* $([S_1] \subseteq [S_2]) \Rightarrow S_2 \subseteq S_1$.

It is to be noted here that the definition of a segment alone does not guarantee disjoint segments. In fact a situation like the following is possible:

Let $S_1$ and $S_2$ be two adjacent segments of $T$ and $\forall x \in S_1 \cap S_2$, $[x] \supseteq [S_1] \cup [S_2]$. It can be easily verified that both $S_1$ and $S_2$ can be valid segments according to the definition. Actually there are two possibilities in this case, depending on the NPS's, that is, either,

$$[S_1] \cap [S_2] = [S_1] \text{ or } [S_2], \quad \text{for } S_1 \subseteq S_2 \text{ or } S_2 \subseteq S_1$$

or

$$[S_1] \cap [S_2] \neq [S_1] \text{ or } [S_2], \quad \text{for } S_1 \cap S_2 \neq \emptyset \text{ or } S_1 \text{ or } S_2.$$

So to ensure a proper partitioning of the structure we need to add another condition that when a point $x$ belongs to an existing segment $S_1$, it cannot be considered a candidate for another segment $S_2$. Eventually $S_1$ and $S_2$ above would be modified as $S_1' = S_1$ and $S_2' = S_2 - S_1$ or $S_2' = S_2$ and $S_1' = S_1 - S_2$, depending on whether $S_1'$ or $S_2'$ is constructed first. Clearly, $S_1' \cap S_2' = \emptyset$. Hence after the construction of one segment $S$ in the structure, only the rest of the structure $T - S$ should be considered for segmentation.

Now in the case when $[S_1] \subseteq [S_2]$, we have an interesting situation. Because, if $S_1'$ is first constructed, we have only one segment (since $S_2' = \emptyset$), whereas if $S_2'$ is first constructed, we have two segments. So from the minimality of the number of segments, $S_1'$ should be constructed first. Moreover, it is also intuitively unacceptable from the structural point of view that a set of points ($S_2'$) having similar neighbouring plane features as its surrounding ($S_1'$) should be marked as a separate segment.

Hence we formulate the following strategy for the generation of segments.

**Strategy.** *If two segments $S_1$ and $S_2$ are adjacent, then it is required that $[S_1] \not\subseteq [S_2]$ and $[S_2] \not\subseteq [S_1]$.*

So out of two possible segments, it is always required to construct the one with the smaller value of $|[S]|$.

The requirement of completeness is however easy to satisfy because till even a single point is left in the structure $T$ which does not belong to any segment, we can mark it as a new segment according to the definition. The details of the algorithm will now be presented in the next section.

## 4. The algorithm

In the previous section we have discussed the segmentation scheme mainly from the point of view of validation. That is, if a structure is already segmented, we can adjudge the quality of the segmentation or in particular whether a better segmentation is possible with fewer segments. However, it always assumes that the value of $[S]$ is known a priori and no method has been given for its selection. So before presenting the algorithm here, we first define the concept of prime basis sets which can be used for the generation of a segment.

A *prime basis set* ($B_p$) of a segment $S$ is defined as the minimal connected subset of $S$ for which $[B_p] = [S]$. So once a prime basis set of a segment is known, the entire segment can be easily constructed. Moreover, for minimal number of segments we require to choose a $B_p$ in the structure which has minimum $|[B_p]|$. This selection is unfortunately, combinatorially explosive. In fact, there are $2^n$ possible prime basis sets in a structure with $n$ points. So the selection of the prime basis set dominates the time complexity and the algorithm turns out to be exponential. A solution to this problem is offered subsequently by the introduction of prime elements in place of prime basis sets. A *prime element* of a segment $S$ is defined as a point $x \in S$ such that $[x] = [S]$. A prime element with minimum $|[x]|$ is then used for constructing the segment. It is to be noted here that a given segment may or may not have a prime element. But it is always possible to define a segment from a prime element.

The algorithm for segmentation is now presented below in terms of the prime basis sets. The variant using the prime elements is a natural extension.

**Algorithm Segmentation**

*Input*: A structure $T$.

*Output*: A set of segments $S_i$, $1 \leqslant i \leqslant m$ of $T$, such that

$$S_i \cap S_j = \emptyset,\ i \neq j, \quad \text{and} \quad \bigcup_{i=1}^{m} S_i = T.$$

1. For all $x \in T$, evaluate $[x]$.
2. Choose the prime basis set $B_p$ of $T$ which has minimum $|[B_p]|$.
3. Define segment $S(B_p)$ from $B_p$ using the definition of a segment.
4. $T \leftarrow T - S(B_p)$.
5. Repeat Steps 2 to 4 until $T = \emptyset$.

(End of Segmentation)

Clearly, the complexity of the algorithm is dominated by Step 2. Hence the following theorem.

**Theorem.** *The worst-case complexity of* Segmentation *is* $O(2^{n+1})$ *if prime basis sets are used and* $O(n^2)$ *if prime elements are used, where* $n = |T|$ = *number of points in a structure.*

It is to be noted that the given algorithm works only for a structure. But an image may have a number of structures $T_1, T_2, \ldots, T_r$ in it. The use of a list of these structures $\{T_i : 1 \leqslant i \leqslant r\}$ in place of the single structure $T$ in the algorithm helps to tackle these situations. Though the use of the prime elements is more attractive from the point of view of computational expense, the use of prime basis sets gives fewer segments. Also in the latter case the segments are structurally more meaningful than in the former. So if the image size is not very large and exponential cost is affordable, it is better to use prime basis sets. The prime elements would however generate a workable solution within short time.

## 5. Results

The algorithm was tested with a large set of synthetic data. The solid objects are first thinned so that the medial surfaces of the objects are obtained. Some of these test objects are shown in Figures 4(a), 5(a) and 6(a). The segmented surfaces are obtained as in Figures 4(b), 5(b) and 6(b), respectively. The respective segments can be labelled by an integer and can also be characterised by their neighboring plane set denoting the orientation of the plane or a line.

During segmentation the information of the adjacent segments of a particular segment and the way in which the segments are connected (characterised by a set of digital neighborhood plane sets) are also obtained. Thus Figure 4(b) can be described as follows: (Note that a segment $S$ is of type $i$ if $[S] = i$.)

Segment 1 is of type 1.
Segment 2 is of type 2.
Segment 2 is connected to segment 1 by the type of 4.
Segment 3 is of type 2.
Segment 3 is connected to segment 1 by the type of 5.
Segment 4 is of type 4.
Segment 4 is connected to segment 1 by the type of 4.
Segment 5 is of type 5.
Segment 5 is connected to segment 1 by the type of 5.

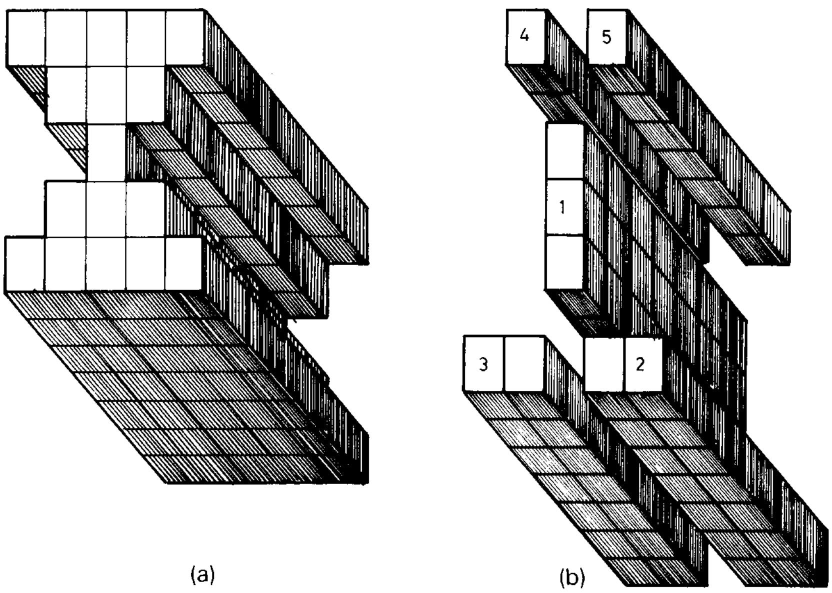


Figure 4. (a) Voxel representation of a solid object. (b) Segments of the thinned surface of the solid object shown in (a).

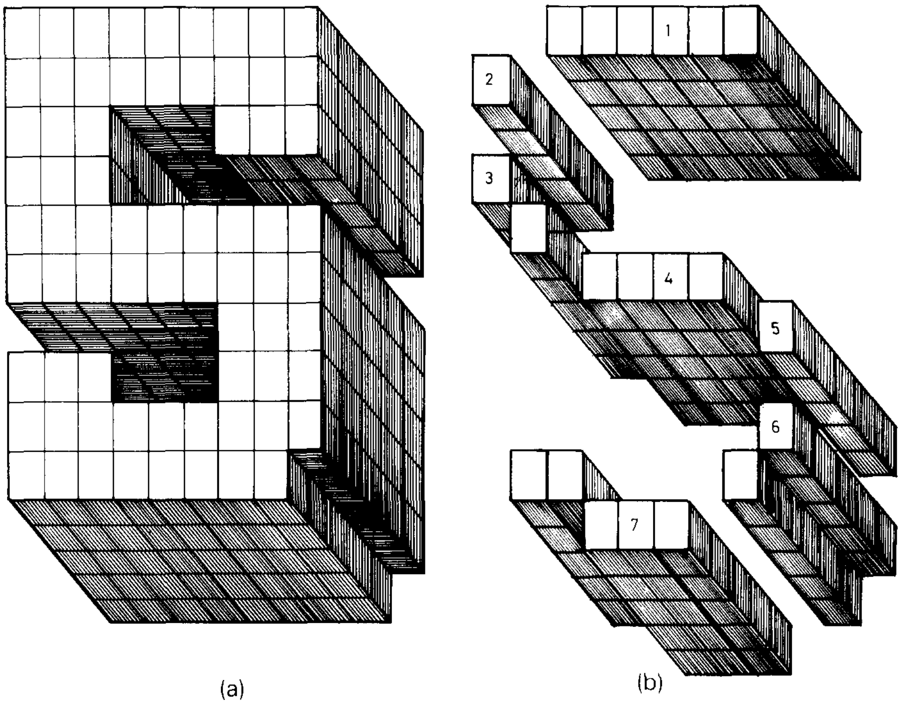


Figure 5. (a) Voxel representation of a 3-D $S$. (b) Segments of the thinned surface of the 3-D $S$.

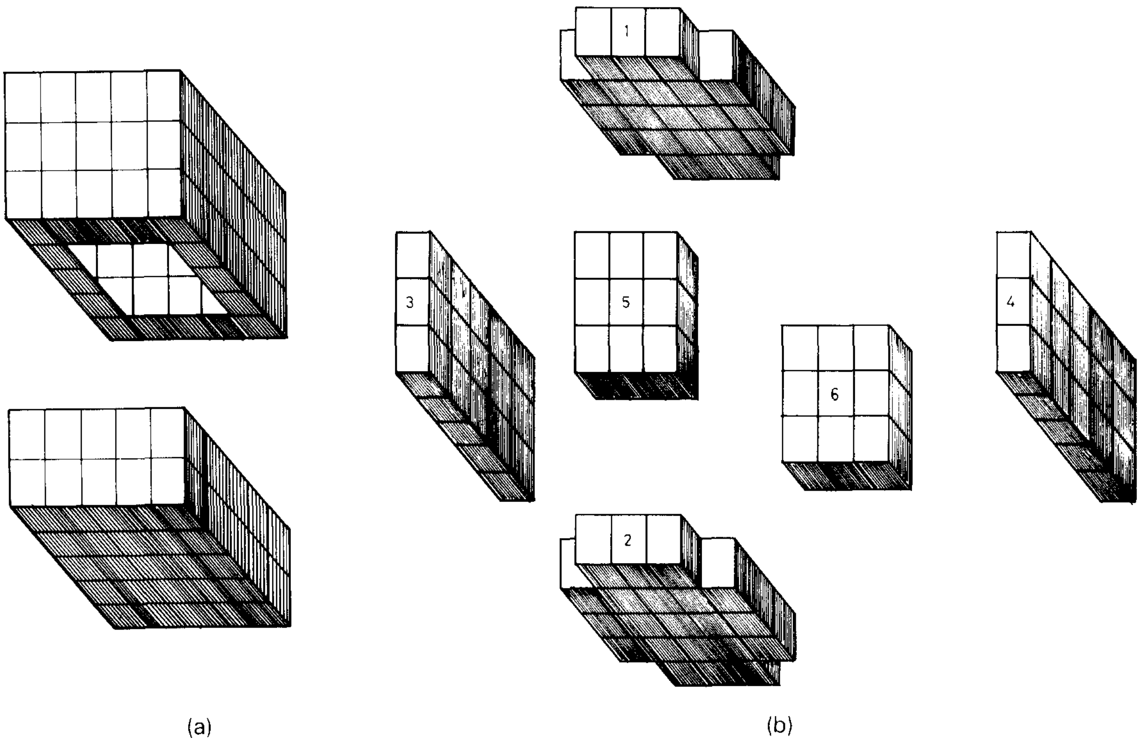


Figure 6. (a) Voxel representation of cube with a hole inside. (b) Segments of the boundary surface of the cube.

In the description given above, the type denotes the characterising neighborhood plane set of the corresponding segments and the connection between two adjacent segments.

In particular Figure 6 illustrates a special situation where the boundary surface has not been thinned before segmentation; the thickness of the surface in this case accounts for non-uniform segments in the result.

It should be noted that while segmenting a structure $T$ we have the flexibility of engineering the threshold value ($k$) so as to optimise our segmentation process. The choice of $k=3$ for thinned surfaces and $k=4$ to 6 for thick surfaces was found to be satisfactory.

## 6. Remarks

The concept of digital neighborhood plane has been used in this paper for the segmentation of 3-D surfaces. The definition of segmentation used here alone does not ensure disjoint segments. However it may be noted that in the step of region growing, if the $[z] \cap [S] = [S]$ is modified to $[z] = [S]$, then segments will be disjoint. But in that case the number of segments will be large and as some experiments show, the generated segments would not retain the structural information well. Hence, the use of the present definition along with the additional criteria is preferrable.

Because of the dominance of planar features in the digital neighbourhood planes, the present segmentation algorithm works better for the polyhedral surfaces than for the curved surfaces. It should be noted that the segmentation technique described in this paper is not invariant under rotation of the objects in 3-D space. However much structural information is obtained which could be used for further analysis.

## Acknowledgement

The first author gratefully acknowledges the financial support received in the form of a senior research fellowship of CSIR (Grant No. 9/81/(71)/86-EMR-I). The authors are thankful to the anonymous referee for the comments on an earlier version of the paper.

## References


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

Boissonnat, J.D. and O.D. Faugeras (1981). Triangulation of 3-D objects. *Proc. 7th Int. Joint Conf. on A.I.*, Vancouver, Canada, Aug. 24–28, 1981. IJACI, New York, 658–660.

Boissonnat, J.D. and O.D. Faugeras (1982). Representation of object triangulation points in 3-D space. *Proc. 6th Int. Conf. on Pattern Recognition*, Munich, West Germany, Oct. 19–22, 1982. IAPR and IEEE, New York, 830–832.

Dane, C. and R. R. Bajcsy (1982). An object centered three dimensional model builder. *Proc. 6th Int. Conf. on Pattern Recognition*, Munich, West Germany, Oct. 19–20, 1982. IEEE, New York, 348–350.

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

Faugeras, O.D. and M. Hebert (1983). A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces. *Proc. 7th Int. Joint Conf. on A.I.*, Vancouver, Canada, Aug. 24–28, 1983. IJACI, New York, 996–1002.

Hebert, M. and J. Ponce (1982). A new method for segmentating 3-D scenes into primitives. *Proc. 6th Int. Conf. on Pattern Recognition*, Munich, West Germany, Oct. 19–22, 1982. IAPR and IEEE, New York, 836–838.

Henderson, T.C. (1982). Efficient segmentation method of range data. *Proc. Soc. Photo-optical Instrumentation Engineers Conf. on Robot Vision, Vol. 336*, Arlington, VA, May 6–7, 1982. SPIE, Bollingham, WA, 46–47.

Henderson, T.C. and B. Bhanu (1982). Three point seed method for the extraction of planar faces from range data. *Proc. Workshop on Industrial Application of Machine Vision*, Research Triangle Park, NC, May 1982. IEEE, New York, 181–186.

Inokuchi, S. and R. Nevatia (1980). Boundary detection in range pictures. *Proc. 5th Int. Conf. on Pattern Recognition*, Miami, FL, Dec. 1–4, 1980. IAPR and IEEE, New York, 1031–1035.

Inokuchi, S., T. Nita, F. Matsuday and Y. Sakurai (1982). A three dimensional edge-region operator for range pictures. *Proc. 6th Int. Conf. on Pattern Recognition*, Munich, West Germany, Oct. 19–22, 1982. IAPR and IEEE, New York, 918–920.

Milgrim, D.L. and C.M. Bjorklund (1980). Range image processing: planar surface extraction. *Proc. 5th Int. Conf. on Pattern Recognition*, Miami, FL, Dec. 1–4, 1980. IEEE, New York, 921–929.

Shapiro, R. and H. Freeman (1977). Recognition of curved surface bodies from a set of imperfect projections. *Proc. 5th Int. Joint Conf. on A.I.*, Cambridge, MA, Aug. 22-25, 1977. IJCAI, New York, 628-634.

Vemuri, B.C. and J.K. Aggarwal (1984). Three dimensional reconstruction of objects from range data. *Proc. 7th Int. Conf. on Pattern Recognition*, Montreal, Canada, July 30-Aug. 2, 1984, 752-754.