Detection of 3-D Simple Points for Topology Preserving Transformations with Application to Thinning

P. K. Saha and B. B. Chaudhuri

***Abstract*— The problems of 3-D digital topology preservation under binary transformations and 3-D object thinning are considered in this correspondence. At first, we establish the conditions under which transformation of an object voxel to a non-object voxel, or its inverse does not affect the image topology. An efficient algorithm to detect a simple point has been proposed on the basis of those conditions. In this connection, some other interesting properties of 3-D digital geometry are also discussed. Using these properties and the simple point detection algorithm, we have proposed an algorithm to generate surface-skeleton so that the topology of the original image is preserved, the shape of the image is maintained as much as possible, and the results are less affected by noise.**

***Index Terms*—3-D digital topology, tunnel, simple point, binary transformation, 3-D thinning, surface-skeleton.**

I. Introduction

This correspondence is concerned with the topology preserving transformations of three-dimensional (3-D) two-tone images and their application to 3-D image thinning. The study of digital topology [2], [3], [9] provides a sound mathematical basis for various image processing operations such as thinning [1], [4], [5], [10], [11], surface recognition, and genus evaluation, etc. In this correspondence, we consider binary voxel representation of a 3-D image and concentrate on the effects of transforming an object voxel to a nonobject voxel (deletion) or its inverse (addition). More specifically, we want to investigate the conditions under which such transformations do not disturb the topology of an image. A voxel satisfying such conditions is called a *simple voxel* or *simple point*. One major aim of our correspondence is to establish a simple point characterization that is easy to compute. The study can be useful in most of the operations stated above.

The general definitions are given in Section II wherefrom voxels are almost always referred to as *points*. The problems related to simple point are addressed in Section III. Also, an efficient algorithm to detect a simple point is developed in this section. Using the development of Section III, schematic description of a 3-D image thinning algorithm is proposed in Section IV. We consider surface-thinning for 3-D images. The algorithm preserves the topology and tries to preserve the shape of the original image as much as possible. Also, the algorithm is less susceptible to noise in the image.

II. General Definitions and Notations

A 3-D digital image is defined in a 3-D array of lattice points $\mathcal{V}$ represented by integer valued triples of Cartesian Co-ordinates (i, j, k). We follow the conventional definition of *α-neighborhood* or *α-adjacency* [2] of points where $\alpha \in \{6, 18, 26\}$. Let S be a nonempty subset of $\mathcal{V}$. An *α-path* of *length* $n > 0$ from p to q in S means a sequence of distinct points $p = p_0, p_1, \cdots, p_n = q$ of S such that p_i is α-adjacent to $p_{i+1}, 0 \leq i < n$. A single point is an α-path of *length* 0. An α-path $p_0, p_1, \cdots, p_n$ is an *α-closed path* iff p_0 is α-adjacent to p_n. Two points $p, q \in S$ are *α-connected* in S iff there exists an α-path from p to q in S. The equivalence classes of S under α-connectivity are called *α-components* of S. A set of points S is called α-connected iff every two points $p, q \in S$ are α-connected in S.

In the following discussions, $\mathcal{N}(p)$ is used to denote the set of 27 points in $3 \times 3 \times 3$ neighborhood of p. With respect to $\mathcal{N}(p)$, we classify the points according to their adjacency relations with p as follows.

s-point: An s-point is 6-adjacent to p.

e-point: An e-point is 18-adjacent but not 6-adjacent to p.

v-point: A v-point is 26-adjacent but not 18-adjacent to p.

Let $p = (l_1, l_2, l_3)$ be a point and let $x = (m_1, m_2, m_3)$ be an s-point of $\mathcal{N}(p)$ i.e., $|m_i - l_i| = 1$ for some i. An *x- surface* of $\mathcal{N}(p)$ is the set of points $(n_1, n_2, n_3) \in \mathcal{N}(p)$ where $n_i = m_i$. Thus, an x-surface contains exactly one s-point. Similarly, let $y = (o_1, o_2, o_3)$ be an e-point of $\mathcal{N}(p)$ i.e., $|o_i - l_i| = 1$ and $|o_j - l_j| = 1$ for some $i \neq j$. A *y edge* of $\mathcal{N}(p)$ is the set of points $(h_1, h_2, h_3) \in \mathcal{N}(p)$ where $h_i = o_i$ and $h_j = o_j$. Thus, a y-edge contains exactly one e-point. The points of $\mathcal{N}(p)$ excluding p is denoted as $\mathcal{Y}(p)$. Thus, $\mathcal{Y}(p)$ is the border of $\mathcal{N}(p)$. Two s-points $a, b \in \mathcal{N}(p)$ are called *opposite* if they are not 18-adjacent. Otherwise, they are called *nonopposite s-points*. Let a, b, c denote three nonopposite s-points in $\mathcal{N}(p)$. Then we define the following functions.

$$f_1(a, b, p) = q | q \in \mathcal{Y}(p) \text{ and 6-adjacent to } a, b.$$

$$f_2(a, b, c, p) = q | q \in \mathcal{Y}(p) \text{ and 6-adjacent to } f_1(a, b, p),$$

$$f_1(b, c, p), f_1(c, a, p).$$

We follow the conventional representation [2] of a 3-D digital image $\mathcal{P}$ by a quadruple $(\mathcal{V}, \alpha, \beta, B)$. The image space $\mathcal{V}$ is the set of all 3-D cubic grid points in a finite rectangular parallelpiped. We consider 26-adjacency for black points and 6-adjacency for white points. A 26-component of B is a *black component* and a 6-component of $\mathcal{V} - B$ is a *white component*. *Border* $\mathcal{Y}$ of an image space $\mathcal{V}$ is defined as the set of points p (called *border points*) such that $\mathcal{N}(p) \not\subset \mathcal{V}$. *Interior* $\mathcal{I}$ of an image space $\mathcal{V}$ is defined as the set of points p (called *interior points*) such that $\mathcal{N}(p) \subset \mathcal{V}$. Obviously, $\mathcal{I} = \mathcal{V} - \mathcal{Y}$.

III. (26, 6) Simple Point: Theory and Algorithm

In this section, we derive an algorithm for efficient detection of (26, 6) simple points. As discussed in [2] a point p is a simple point iff its deletion (or addition) preserves 3-D topology in the current black and white configuration of $\mathcal{N}(p)$. The object points in an image may be grouped as a set of connected components. In general, a component may contain cavities and tunnels. A *cavity* is a 3-D analog of a hole in 2-D whereby nonobject points generate a component surrounded by an object component. A *tunnel*, on the other hand, does not generate a new nonobject component. However, an object component contains a tunnel iff it generates a solid handle or it contains a hollow torus [3]. An example is the hollow cylinder. In 2-D there is no concept analogous to 3-D tunnel. The numbers of object components, tunnels and cavities in an image denote its 0th, 1st, and 2nd Betti numbers, respectively. Thus, a point p is a simple point iff its binary transformation does not change these three numbers in $\mathcal{N}(p)$.

Let $\mathcal{P} = (\mathcal{V}, 26, 6, B)$ be the image under consideration. To detect p as a simple point in $\mathcal{P}$, we define two subimages in $\mathcal{N}(p)$ as follows:

$$\hat{\mathcal{N}}(p) = (\mathcal{N}(p), 26, 6, (\mathcal{N}(p) \cap B) \cup \{p\})$$
$$\check{\mathcal{N}}(p) = (\mathcal{N}(p), 26, 6, (\mathcal{N}(p) \cap B) - \{p\}).$$

In other words, p is always white in $\check{\mathcal{N}}(p)$ (i.e., p is deleted) while p is always black in $\hat{\mathcal{N}}(p)$ (i.e., p is added). At any other position of $\mathcal{N}(p)$, the color in $\hat{\mathcal{N}}(p)$ or $\check{\mathcal{N}}(p)$ is the same as that of $\mathcal{P}$. Thus, the definition of simple point can be stated as: a point $p \in V$ is a simple point iff $\check{\mathcal{N}}(p)$ and $\hat{\mathcal{N}}(p)$ are topologically equivalent. Before discussing topological equivalence we develop a formal condition for the existence of tunnels.

A. Existence of Tunnel in (26, 6) Connectivity

As defined in [3], an image contains tunnel iff there exists a solid handle or a hollow torus. Now, if the set of black points generate a solid handle or a hollow torus then at least one interior point must be white. Thus, an image $\mathcal{P} = (\mathcal{V}, 26, 6, B)$ where $\mathcal{I} \subset B$ never contains a tunnel ($\hat{\mathcal{N}}(p)$ belongs to this class). We now define the existence of tunnel in the class of images $\mathcal{P} = (\mathcal{V}, 26, 6, B)$ where $B \subset \mathcal{Y}$ ($\check{\mathcal{N}}(p)$ belongs to this class). Let $\mathcal{P}' = (\mathcal{V}, 26, 6, B')$ be a shrunk version [2] of $\mathcal{P}$. Then $\mathcal{P}$ contains a tunnel iff $\mathcal{P}'$ contains no cavity (i.e., $\exists p \in \mathcal{Y} - B'$ and p is 6-adjacent to $\mathcal{I}$) and there exists a nontrivial 26-closed path (*length* > 0) in B'.

Although formal definition for the existence of tunnel in a restricted class of images is proposed above, we shall derive a simpler condition for the existence of tunnel in $\check{\mathcal{N}}(p)$. At first, we introduce the concept of *crossing*. Consider an image $\mathcal{P} = (\mathcal{V}, 26, 6, B)$ where $B \subset \mathcal{Y}$. Then, a 6-path $\pi \subset \mathcal{Y} - B$ crosses a 26-path in B at a point $p \in \pi$ iff $\mathcal{N}(p) \cap \mathcal{Y} - \pi$ is 26-connected and two 6-components of $\mathcal{N}(p) \cap \mathcal{Y} - \pi$ intersect with B (thus $\mathcal{N}(p) \cap \mathcal{Y} - \pi$ is not 6-connected). An example of crossing is demonstrated in Fig. 1 where a 6-path of white border points (shown by dotted cubes) crosses a 26-path of black points (shown by shaded cubes with bold outlines) at p. If crossing is allowed then an anomalous situation may occur where the image contains a tunnel on the border and the set of white border points is 6-connected as shown in Fig. 1 (so, we cannot have a parallel result to the Jordan curve theorem). Consider the following notations:

$\mathcal{Y}_B(p) = \mathcal{Y}(p) \cap B$; ($\mathcal{Y}_B(p)$ denotes the set of black points in $\check{\mathcal{N}}(p)$)

$\mathcal{Y}_W(p) = \mathcal{Y}(p) - \mathcal{Y}_B(p)$;

$W_s(p) =$ the set of white s-points in $\check{\mathcal{N}}(p)$;

$W_e(p) =$ the set of white e-points in $\check{\mathcal{N}}(p)$;

$W_{se}(p) = W_s(p) \cup W_e(p)$.

Using the definition for the existence of tunnel and the definition of crossing we arrive at the following observations which are summarized in Observation 1.

- Consider Fig. 2. Here, the set of black points belongs to the border and constitutes a tunnel. Note that the set of black points generates a 26-closed path. Consider two 6-connected subsets S_1 and S_2 of white border points as shown in the figure. They are 6-connected to each other by interior points. But it is not possible to connect them by a 6-path of white border points without crossing the 26-path of black points.
- More specifically, we may observe in $\check{\mathcal{N}}(p)$ that p is the only interior point and $\mathcal{Y}_B(p)$ belongs to the border of $\mathcal{N}(p)$. Now, a tunnel exists in $\check{\mathcal{N}}(p)$ iff there exist two 6-connected subsets $S_1, S_2 \subset \mathcal{Y}_W(p)$ such that S_1 and S_2 are 6-adjacent to p but they are not connected by a 6-path in $\mathcal{Y}_W(p)$ without crossing any 26-path in $\mathcal{Y}_B(p)$. Finally, a 6-connected subset of $\mathcal{Y}_W(p)$ is 6-adjacent to p iff it contains one or more s-points.

Observation 1: $\check{\mathcal{N}}(p)$ contains no tunnel iff every two points of $W_s(p)$ are connected by a 6-path in $\mathcal{Y}_W(p)$ nowhere crossing a 26-path in $\mathcal{Y}_B(p)$.

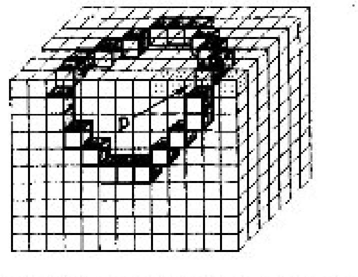

Fig. 1. Demonstration of crossing. (Here, the shaded cubes with bold outlines are black points. All other cubes are white points. A 6-path of white points shown by dotted cubes crosses the 26-path of black points at p.)

Theorem 1: $\check{\mathcal{N}}(p)$ contains no tunnel iff $W_s(p)$ is 6-connected in $W_{se}(p)$.

Proof: According to Observation 1, we have to establish the following fact. Two points of $W_s(p)$ are connected by a 6-path in $\mathcal{Y}_W(p)$ nowhere crossing a 26-path in $\mathcal{Y}_B(p)$ iff they are 6-connected in $W_{se}(p)$. Here, we use $(a, d), (b, e), (c, f)$ to denote three distinct unordered pairs of opposite s-points in $\mathcal{N}(p)$.

Consider a 6-path π in $\mathcal{Y}_W(p)$ between two points $s_1, s_2 \in W_s(p)$ which does not cross a 26-path in $\mathcal{Y}_B(p)$ and $\pi \cap (\mathcal{Y}_W(p) - W_{se}(p)) \neq \phi$ (i.e., π contains some v-points). We find a 6-path π' in $W_{se}(p)$ from s_1 to s_2. Let π be the 6-path $p_0, \cdots, p_i, f_2(a, b, c, p), p_{i+1}, \cdots, p_n$. It follows from $\mathcal{N}(p)$ that $f_2(a, b, c, p)$ is 6-adjacent to $f_1(a, b, p)$, $f_1(b, c, p)$ and $f_1(c, a, p)$ only. Let $p_i = f_1(a, b, p)$ and $p_{i+1} = f_1(b, c, p)$. Since π does not cross a 26-path in $\mathcal{Y}_B(p)$, either $b \in \mathcal{Y}_W(p)$ or $\{a, c, f_1(c, a, p)\} \subset \mathcal{Y}_W(p)$. In either case $f_1(a, b, p)$ is 6-connected to $f_1(b, c, p)$ in $W_{se}(p)$. Replacing all such v-points in π we can get a 6-path π' in $W_{se}(p)$ from s_1 to s_2. Hence, every two points of $W_s(p)$ are connected by a 6-path in $\mathcal{Y}_W(p)$ nowhere crossing a 26-path in $\mathcal{Y}_B(p)$ implies that $W_s(p)$ is 6-connected in $W_{se}(p)$.

Now we show that a 6-path π in $W_{se}(p)$ between two points $s_1, s_2 \in W_s(p)$ never crosses a 26-path in $\mathcal{Y}_B(p)$. Let $f_1(a, b, p)$ be an e-point belonging to π. Then π contains the sequence "$\cdots, a, f_1(a, b, p), b, \cdots$." Here, $\mathcal{N}(f_1(a, b, p)) \cap \mathcal{Y}(p) - \pi$ is not 26-connected. Thus, π never crosses a 26-path of $\mathcal{Y}_B(p)$ at an e-point. Now, consider an s-point $a \in \pi$. Then, π contains one of the following four sequences:

Class 1) $a \Rightarrow \pi$ contains only a;
Class 2) $a, f_1(a, b, p), b, \cdots \Rightarrow a$ is an end point of π;
Class 3) $\cdots, b, f_1(a, b, p), a, f_1(a, c, p), c, \cdots \Rightarrow b, c$ are nonopposite;
Class 4) $\cdots, b, f_1(a, b, p), a, f_1(a, e, p), e, \cdots \Rightarrow b, e$ are opposite.

In the first two cases $\mathcal{N}(a) \cap \mathcal{Y}(p) - \pi$ is 6-connected while $\mathcal{N}(a) \cap \mathcal{Y}(p) - \pi$ is not 26-connected in last two cases. Thus, π never crosses a 26-path of $\mathcal{Y}_B(p)$ at an s-point. □

Corollary 1: Existence of tunnel in $\check{\mathcal{N}}(p)$ is independent of the color of v-points.

B. Characterization of (26, 6) Simple Point

Our characterization of (26, 6) simple point is based on the equivalence between $\check{\mathcal{N}}(p)$ and $\hat{\mathcal{N}}(p)$ with respect to numbers of black components, tunnels and cavities. We have the following observations on $\hat{\mathcal{N}}(p)$.

Class 1) $\hat{\mathcal{N}}(p)$ contains no cavity (since, p is the only interior point which is black).
Class 2) $\hat{\mathcal{N}}(p)$ contains exactly one black component (since, every point of $\mathcal{N}(p)$ is 26-adjacent to p which is black).
Class 3) $\hat{\mathcal{N}}(p)$ contains no tunnel (see the previous subsection).

Thus, a point $p \in V$ is a simple point iff $\hat{N}(p)$ and $\hat{N}'(p)$ are topologically equivalent i.e., iff all the following conditions are satisfied on $\hat{N}(p)$.

Condition 1a: $\hat{N}(p)$ contains no cavity.

Condition 1b: $\hat{N}(p)$ contains at least one black point.

Condition 1c: The black points of $\hat{N}(p)$ are 26-connected.

Condition 1d: $\hat{N}(p)$ contains no tunnel.

$\hat{N}(p)$ contains no cavity iff at least one s-point is white. Conditions 1b–c need no more explanation while the existence of tunnel in $\hat{N}(p)$ is defined in Theorem 1. This characterization for (26, 6) simple point was earlier proposed in [7], [6].

C. Implementation Approach

Based on the characterization of (26, 6) simple point established in the previous subsection, we develop an algorithm to detect (26, 6) simple point.

Observation 2: Let y be a point belonging to x-surface and q be another point in $\hat{N}(p)$. Then q is 26-adjacent to y implies that q is 26-adjacent to x.

Proposition 1: Let x be a black s-point in $\hat{N}(p)$. Then connectedness of black points of $\hat{N}(p)$ is independent of the color of other points of x-surface.

Proof: Consider a point $y \in x$-surface and $y \neq x$. Let S be the set of black points in $\hat{N}(p)$. Every two points of $S - \{y\}$ which are 26-connected in $S - \{y\}$ are also 26-connected in $S \cup \{y\}$ and y is 26-adjacent to $x \in S - \{y\}$. Thus, $S - \{y\}$ is 26-connected implies that $S \cup \{y\}$ is 26-connected. Let us consider two points $q, r \in S - \{y\}$ such that they are 26-connected in $S \cup \{y\}$ but not 26-connected in $S - \{y\}$. Then a 26-path $q = p_0, \cdots, p_i, y, p_{i+1}, \cdots, p_n = r$ exists from q to r in $S \cup \{y\}$. By Observation 2, p_i and p_{i+1} are 26-adjacent to x. Hence, the 26-path $q = p_0, \cdots, p_i, x, p_{i+1}, \cdots, p_n = r$ exists from q to r in $S - \{y\}$—Contradiction!! Hence, $S \cup \{y\}$ is 26-connected implies that $S - \{y\}$ is 26-connected. □

Proposition 2: If an s-point x is black in $\hat{N}(p)$ then existence of tunnel in $\hat{N}(p)$ is independent of the color of other points of x-surface.

Proof: Let S be the set of white s-points and S' be the set of white s-points and e-points in $\hat{N}(p)$. By Theorem 1, $\hat{N}(p)$ contains no tunnel iff S is 6-connected in S'. Let $y \in x$-surface be an e-point. Clearly, S is 6-connected in $S' - \{y\}$ implies that S is 6-connected in $S' \cup \{y\}$. In $\hat{N}(p)$ no two s-points are 6-adjacent. Also, no two e-points are 6-adjacent. Thus, a 6-path of s-points and e-points is an alternating sequence of s-points and e-points. Hence, a 6-path of $S' \cup \{y\}$ between two points of S contains y implies that $S' \cup \{y\}$ contains two s-points 6-adjacent to y. Now, $\hat{N}(p)$ contains exactly two s-points 6-adjacent to y. Since $y \in x$-surface, x is 6-adjacent to y and by assumption $x \notin S' \cup \{y\}$. Thus, no two points of S are 6-connected in $S' \cup \{y\}$ through y. Hence, S is 6-connected in $S' \cup \{y\}$ implies that S is 6-connected in $S' - \{y\}$. □

As discussed in Proposition 1 and Proposition 2, if an s-point x is black in $\hat{N}(p)$ then connectedness of black points and existence of tunnel in $\hat{N}(p)$ are independent of the color of other points of x-surface. Moreover, the question whether Conditions 1a–b are satisfied in $\hat{N}(p)$, is also independent of the color of other points of x-surface. Thus, we define an x-surface as a *dead-surface* of $\hat{N}(p)$ iff x is black in $\hat{N}(p)$. A v-point or an e-point is called an *effective point* in $\hat{N}(p)$ iff it does not belong to any dead-surface of $\hat{N}(p)$.

Corollary 2: With a known s-point configuration in $\hat{N}(p)$, the effective point configuration of $\hat{N}(p)$ determines whether p is a simple point.

Corollary 3: With a known s-point configuration in $\hat{N}(p)$, the effective e-point configuration of $\hat{N}(p)$ determines whether $\hat{N}(p)$ contains a tunnel.

Now, we describe the possible geometric classes of s-point configuration. Two configurations belong to the same geometric class iff one can be transformed to the other by three-dimensional rotation in multiples of 90° about different axes (with p as origin). Possible geometric classes of s-point configuration and corresponding number of effective points (n_e) are as follows.

- Class 0) Six s points are black ($n_e = 0$).
- Class 1) Five s-points are black ($n_e = 0$).
- Class 2) Two pairs of opposite s-points are black ($n_e = 0$).
- Class 3) Two opposite s-points and two nonopposite s-points are black ($n_e = 1$).
- Class 4) Two opposite s-points and another s-point are black ($n_e = 2$).
- Class 5) Three nonopposite s-points are black ($n_e = 4$).
- Class 6) Two opposite s-points are black ($n_e = 4$).
- Class 7) Two nonopposite s-points are black ($n_e = 7$).
- Class 8) One s-point is black ($n_e = 12$).
- Class 9) No s point is black ($n_e = 20$).

As discussed earlier, the configuration of n_e effective points determines whether p is a simple point. Thus, if $n_e = 0$ as in Classes 0–2, we at once know whether p is a simple point or not. For Class 0 and Class 2 p is never a simple point while p is always a simple point for Class 1. In other cases as in Classes 3–9 where $n_e > 0$, we use a look_up_table to determine whether p is a simple point. For a given s-point configuration there are 2^{n_e} possible effective point configurations. An effective point configuration is a n_e-bit binary number. For example, let $n_e = 4$ as in Class 5 with e_1, e_2, e_3, e_4 denoting the four effective points. Then a 4-bit binary number is generated such that its ith bit denotes the color of e_i (i.e., "1" when e_i is black and "0" otherwise). For each such configuration of effective points there is an entry in the look_up_table which contains a 0–1 (one bit) flag to denote if p is a simple point. Actual ordering of effective points depends on the specific s-point configuration of the class. We denote the ordering of effective points for an s-point configuration γ as $EFO(\gamma)$. Let us consider that we have a look_up_table for the s-point configuration γ with the effective point ordering as $EFO(\gamma)$. Let γ' denote another s-point configuration in the same geometric class. Let $R_{\gamma,\gamma'}$ denote the 3-D rotation that transforms γ to γ'. Then the same look_up_table can be used if we make the effective point ordering as $EFO(\gamma')$ such that its ith point is obtained by applying the rotation $R_{\gamma,\gamma'}$ to the ith point of $EFO(\gamma)$. Hence, only one look_up_table is necessary for each class. Let LT_i denote the look_up_table for Class i. Thus, the total storage requirement for the look_up_tables of Class $i, 3 \leq i \leq 8$ is: $2^1 + 2^2 + 2^4 + 2^4 + 2^7 + 2^{12}$ bits = 0.5 Kbytes (considering one bit for each entry). Unfortunately, the memory space required for the look_up_table of Class 9 is as high as 2^{20} bits i.e., 128 Kbytes. Hence, a straight-forward use of look_up_table just described is not efficient for Class 9. We use the following development to generate a different look_up_table for Class 9 that needs less space.

Observation 3: Let the point $y \in x$-edge and q be another point in $\hat{N}(p)$. Under this assumption q is 26-adjacent to y implies that q is 26-adjacent to x.

Proposition 3: If an e-point x is black in $\hat{N}(p)$ then whether p is a simple point or not, is independent of the color of the other points of x-edge.

Proof of Proposition 3 is similar to that of Proposition 1. We define an x-edge as a *dead-edge* in $\hat{N}(p)$ iff x is black. A v-point is an *isolated point* in $\hat{N}(p)$ iff it neither belongs to any dead-surface nor belongs to any dead-edge.

Corollary 4: Let y be a black isolated point in $\hat{N}(p)$. Then $\{y\}$ is a black component of $\hat{N}(p)$.

When all s-points are white, there may be two cases as follows.

Case 1: All e-points are white.

Here, all v-points in $\hat{N}(p)$ are isolated points and $\hat{N}(p)$ never contains a tunnel. Hence, p is a simple point iff exactly one v-point is black.

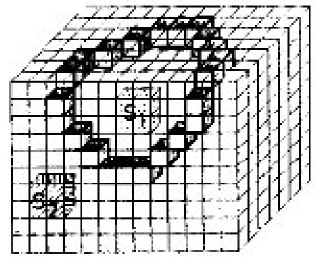

Fig. 2. Examples of tunnel when all black points lie on the border. (Black points are shown as in Fig. 1. See text for other explanations.)

Fig. 3. Results of sequential thinning. From left to right: original shape; surface-skeleton; original shape with noise; surface-skeleton.

Case 2: At least one e-point is black.

Here, p is not a simple point if any of the following situations occurs—1) the set of black e-points is not 26-connected, 2) there is a tunnel in $\hat{N}(p)$, or 3) there is a black isolated point.

At first we try to find a black e-point in $\hat{N}(p)$. If an e-point x is black then there can be at most 2^{11} possible configurations of other e-points and hence the look_up_table LT_5 contains 2^{11} entries. An ordered set $EEO(x)$ of these eleven e-points is used to calculate their configuration value. The address of an entry of LT_5 corresponds to a distinct configuration of $EEO(x)$. At each entry of LT_5 we store the following information—1) whether the set of black e-points contains more than one 26-component or if there exists a tunnel in $\hat{N}(p)$, and 2) the set of isolated points. Since x is black at most six v-points can be isolated points. We call the ordered set of these six v-points as $ISO(x)$. For each entry of LT_5 we use a one byte word l in which (0–5) bit positions denote the set of isolated points in $ISO(x)$. The 6th bit position of l is "1" if the set of black e-points contains more than one 26-component or if there is a tunnel. The 7th bit position always contains a "0". To determine whether p is a simple point we generate a one byte word w whose (0–5) bit positions denote the configuration of $ISO(x)$ while 6th and 7th bits are "1" and "0", respectively. Thus, p is a simple point iff the *bitwise AND* operation between l and w leads to zero. The look_up_table LT_5 now needs only 2^{11} bytes i.e., 2 Kbytes, instead of 128 Kbytes in its earlier form.

IV. Application to Thinning

Here, we present an application of simple point characterization to 3-D object thinning. A 3-D object can be thinned down to a surface representation of one point thickness. Our surface-thinning is an iterative procedure with two steps namely *primary-surface-thinning* and *final-surface thinning*. In primary-surface-thinning, iterations are continued till the results of current iteration and previous iteration are the same. The output of this step is called *primary-surface-skeleton*. Final surface-thinning is a single iteration procedure producing *final-surface skeleton*. Here, we present a schematic description of the thinning algorithm. For different definitions and explanations see [8].

During each iteration of primary-surface-thinning the set of *open points* [8] is considered for processing. Throughout an iteration, we implicitly memorize two images. One image denotes the black/white configuration at the beginning of current iteration while the other denotes the current state of the processed image. The set of open points is defined on the black/white configuration before current iteration while simple points are always detected on the current configuration. We consider three types of open points, namely *s-open points*, *e-open points*, and *v-open points* [8]. At the beginning of surface-thinning, all black points are initiated as "unmarked" points. As the iterations continue, some of the black points are "marked." Each iteration is completed in three scans of the image space. During the first scan an "unmarked" s-open point is "marked" (such a point is never deleted in subsequent iterations) if it is a *surface point* [8]. Otherwise, it is deleted if it is a simple point. During the second scan an "unmarked" e-open point is deleted if it is a simple point and satisfies some additional constrained described in [8]. During the third scan an "unmarked" v-open point is deleted if it is a simple point. Final-surface-thinning consists of single scan. During this scan any black point (irrespective of whether it is "marked" or "unmarked") is deleted if it is a *surface-erodable point* [8].

To test the effectiveness of the proposed algorithm several 3-D objects were synthesized and subjected to thinning. One of the results is shown in Fig. 3. It is noted that the thinned version preserves the topology and represents the shape of the original image quite faithfully. The skeletal version of the noisy image suggests that the image is quite immune to noise. The thinning algorithm is explained in more detail in [8] where a parallel version of the algorithm is also proposed.

V. Conclusion

We have studied the 3-D digital topology in connection with binary transformation and established a characterization for (26, 6) simple point. Application of the development to 3-D thinning is briefly demonstrated. This study can also be used in other applications such as: recognition of simple surface-point, topology based classification of points in a skeleton , etc. In addition, the development presented here is an advancement to the theory of digital image algebra. Our future research interest involves topology based segmentation of a 3-D object from its skeleton and decomposition of a 3-D object into feature elements from a feature set.

Acknowledgment

The authors wish to thank Prof. D. Dutta Majumder and Prof. A. K. Dutta for their interest in the work.

References

[1] K. J. Hafford and K. Preston, Jr., "Three-dimensional skeletonization of elongated solids," *Comput. Vision Graphics Image Processing*, vol. 27, pp. 78–91, 1984.

[2] T. Y. Kong and A. Rosenfeld, "Digital Topology: Introduction and survey," *Comput. Vision Graphics Image Process.*, vol. 48, pp. 357–393, 1989.

[3] T. Y. Kong, A. W. Roscoe, and A. Rosenfeld, "Concepts of digital topology," *Topology and its Applications*, vol. 46, pp. 219–262, 1992.

[4] S. Lobregt, P. W. Verbeek, and F. C. A. Groen, "Three dimensional skeletonization, principle, and algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 75–77, 1980.

[5] D. G. Morgenthaler, "Three-dimensional Simple points: Serial erosion, parallel thinning and skeletonization," TR-1005, Computer Vision Lab., Univ. of Maryland, 1981.

[6] P. K. Saha, B. B. Chaudhuri, B. Chanda, and D. D. Majumder, "Topology preservation in 3-D digital space," *Pattern Recognition* (accepted).

[7] P. K. Saha, B. Chanda, and D. D. Majumder, "Principles and algorithms for 2-D and 3-D shrinking," TR/KBCS/2/91, N.C.K.B.C.S. Library, Indian Statistical Inst., Calcutta, India, 1991.

[8] P. K. Saha and B. B. Chaudhuri, "Simple point computation and 3-D thinning with parallel implementation," TR/KBCS/1/93, N.C.K.B.C.S. Library, Indian Statistical Inst., Calcutta, India, 1993.

[9] J. I. Toriwaki, S. Yokoi, T. Yonekura, and T. Fukumura, "Topological properties and topology-preserving transformation of a three-dimensional binary picture," in *Proc. 6th Int. Conf. Pattern Recogn.*, pp. 414–419, 1982.
[10] Y. F. Tsao and K. S. Fu, "A parallel thinning algorithm for 3-D pictures," *Comput. Graphics Image Processing*, vol. 17, pp. 315–331, 1981.
[11] Y. F. Tsao and K. S. Fu, "A 3-D parallel skeletonwise thinning algorithm," in *Proc., IEEE PRIP Conf.*, 1982, pp. 678–683.

A Minimum Description Length Model for Recognizing Objects with Variable Appearances (The VAPOR Model)

John Canning

***Abstract*— Most object recognition systems can only model objects composed of rigid pieces whose appearance depends only on lighting and viewpoint. Many real world objects, however, have variable appearances because they are flexible and/or have a variable number of parts. These objects cannot be easily modeled using current techniques. We propose the use of a knowledge representation called the *VAPOR* (Variable APpearance Object Representation) model to represent objects with these kinds of variable appearances. The VAPOR model is an idealization of the object; all instances of the model in an image are variations from the ideal appearance. The variations are evaluated by the *description length* of the data given the model, i.e., the number of information-theoretic bits needed to represent the model and the deviations of the data from the ideal appearance. The shortest length model is chosen as the best description. We demonstrate how the VAPOR model performs in a simple domain of circles and polygons and in the complex domain of finding cloverleaf interchanges in aerial images of roads.**

***Index Terms*— Variable appearance objects, object models, minimum description length, object recognition, shape recognition, knowledge representation.**

I. Introduction

Current object recognition systems can only recognize a limited class of objects. Most systems exploit the viewpoint consistency constraint [9] using knowledge of the number, properties and relationships of the features in the models. For many objects, however, the number, properties and relationships of the features are not fixed.

Throughout this correspondence, we will use the model for a polygon as an example of an object with variable appearance. While the appearance of a particular polygon doesn't vary, the appearance of members of the class of polygons does. An ideal polygon appears as a set of 3 or more non-intersecting, straight edge segments where each endpoint of each segment is coincident with exactly one other endpoint in the set and the chain of edge segments forms a single cycle. Common or expected variations occur in the number, position and orientation of the segments. Unexpected variations occur if there is a different number of cycles in the graph of edge segments or if the endpoints are not exactly coincident, such as in the case of the subjective polygon in Fig. 1.

Kanizsa showed that humans perceive a *subjective contour* of a triangle that occludes parts of the circles (the wedges) in a similar figure [6].

The definition of an object as a variation from an ideal prototype is the basis for our object representation. Instead of modeling an object as a group of lower level features with fixed (or parameterized) spatial relationships, we will model an object as a set of parts with an associated set of constraints. The constraints embody the class of variable appearances that are allowed for the object. All models can be "matched" to a given set of data, but as more constraints are violated, the cost of the match will increase. Since we are concerned only with the appearance of these objects in images, we call our model the VAPOR model (Variable APpearance Object Representation) to signify that it does not represent other aspects of the object such as it's function or composition.

Several methods have been proposed for recognizing objects composed of two or more rigid pieces with parameterized relations between the pieces. Grimson describes a method for objects with a known polygonal silhouette in [5]. Kriegman and Ponce developed a system for recognizing parameterized 3-D objects with curved surfaces [8]. Segen models nonrigid objects with probabilistic hypergraphs [13]. Segen's system learns a class model from a group of training images, and can effectively adjust the hypergraph probabilities to allow object recognition that is invariant to some nonrigid shape variations. None of these systems, however, would be effective for objects with a variable number of parts, and only Segen's can deal with nonrigid shapes. Brooks' ACRONYM system [2] deals explicitly with objects having variable number of parts but does not have a way of comparing competing interpretations.

II. The VAPOR Model—Variable APpearance Object Representation

Even though the objects of interest have variable appearance, we expect to be able to decompose the object into parts. The parts, in turn, can be represented by the same kind of model, leading to a recursive, hierarchical model. Our VAPOR model for a polygon is a *set* of other VAPOR instances, straight edge segments. To find a good instance of a polygon, we must search for a set of segments that satisfy the model constraints. Instead of satisfying hard constraints in the presence of noise, occlusion, etc., it is better to search for a set that is minimally distant from an ideal polygon. We perform this *optimization* by perturbing the set of component instances (edge segments) in order to find a minimum of an energy function. The VAPOR instance perturbation proceeds by adding elements or deleting elements from the set. The VAPOR model is a variation on the idea of the traditional active contour or surface [7], [14] where instead of a fixed size set of sample points, we have a variable size set of other VAPOR models.

Since a VAPOR model does not always correspond to a physical boundary in the way active surfaces do, it is not appropriate to use the analogy of a sheet deforming in a viscous fluid to motivate the optimization. Instead, we minimize an energy function that estimates the variation between the ideal appearance and the appearance of the particular instance.

A. VAPOR Instance Optimization

The optimization of a set is really a *search* among the 2^N possible subsets of the set of N possible elements. The exponential size of this search space means that exhaustive search will be prohibitively expensive when N is large because our computational resources are limited. Hence, we need to use knowledge of the problem domain to reduce the size of the search space and/or focus the search on the shortest path to the goal. If a solution that is slightly sub-optimal is still acceptable, we can develop an algorithm that always returns a solution within a time limit, and as the time limit goes to infinity, the solution approaches the optimal solution. Boddy and Dean call