美国摄影光学仪器工程师学会—国际光学工程学会会议录
第435卷 数字图象处理的计算机系统结构的算法

PROCEEDINGS
Of SPIE-The International Society for Optical Engineering

SPIE

Volume 435

Architecture and Algorithms for Digital Image Processing

André Oosterlinck, Per-Erik Danielsson
Chairmen/Editors

August 25-26, 1983
San Diego, California

Proceedings of SPIE—The International Society for Optical Engineering

Volume 435

Architecture and Algorithms for Digital Image Processing

André Oosterlinck, Per-Erik Danielsson
Chairmen/Editors

August 25-26, 1983
San Diego, California

Published by
SPIE—The International Society for Optical Engineering
P.O. Box 10, Bellingham, Washington 98227-0010 USA
Telephone 206/676-3290 (Pacific Time) • Telex 46-7053

The papers appearing in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and are published as presented and without change, in the interests of timely dissemination. Their inclusion in this publication does not necessarily constitute endorsement by the editors or by SPIE.

Please use the following format to cite material from this book:
Author(s), "Title of Paper," *Architecture and Algorithms for Digital Image Processing,* André Oosterlinck, Per-Erik Danielsson, Editors, Proc. SPIE 435, page numbers (1983).

Library of Congress Catalog Card No. 83-051019
ISBN 0-89252-470-7

Printed in the United States of America.

ARCHITECTURE AND ALGORITHMS FOR DIGITAL IMAGE PROCESSING

Volume 435

Conference Committee

Chairmen
André Oosterlinck
Catholic University of Leuven (Belgium)
Per-Erik Danielsson
Linköping University (Sweden)

Co-Chairmen
Larry Davis
University of Maryland

Steven L. Tanimoto
University of Washington

Jun-ichiro Toriwaki
Nagoya University (Japan)

Session Chairmen
Session 1—New Algorithms
Steven L. Tanimoto, University of Washington

Session 2—Computational Models
Jun-ichiro Toriwaki, Nagoya University, Japan

Session 3—New Architectures
Per-Erik Danielsson, Linköping University, Sweden

Session 4—Architectures and Applications
Larry Davis, University of Maryland

ARCHITECTURE AND ALGORITHMS FOR DIGITAL IMAGE PROCESSING

Volume 435

Contents

ARCHITECTURE AND ALGORITHMS
FOR DIGITAL IMAGE PROCESSING

Volume 435

Session 1

New Algorithms

Chairman
Steven L. Tanimoto
University of Washington

ALGORITHMS FOR SKELETONIZING THREE-DIMENSIONAL DIGITIZED BINARY PICTURES

Jun-ichiro Toriwaki and Shigeki Yokoi

Department of Electronics and Department of Information Science,
Faculty of Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464 Japan

Abstract

In this paper several algorithms for skeletonizing a three-dimensional (3-D) pictorial data are proposed with experimental results to provide ideas on what kinds of operations are required and what the computation time amounts to for processing 3-D images. Algorithms discussed here include shrinking, thinning, distance transformation and border following.

1. Introduction

Recent progress in generating three-dimensional (3D) pictorial data such as CT images and a set of microscopic images strongly requires computer analysis and recognition of 3D digitized pictures. However, algorithm resources available for 3D images is extremely limited compared to the enormous methods which have been developed for two-dimensional image processing. In this paper several algorithms for skeletonizing a three-dimensional (3-D) binary image are proposed with experimental results to show what kinds of operations are required and what the computation time amounts to for recognizing and interpreting 3D images. Algorithms discussed here include shrinking, thinning, distance transformation, and border following, most of which were developed by authors and their colleagues.

2. Definitions and basic properties

A digitized 3-D picture defined on a 3-D square grid is a stack of 2-D digitized images and theoretically modelled as a 3-D array of small cubes called voxels. A voxel located in the i-th row and the j-th column of the k-th plane is denoted by $X = (i,j,k)$. In this paper only binary pictures with K planes including I rows and J columns are considered. (Fig. 1) Voxels with values 0 and 1 are called 0-voxels and 1-voxels, respectively. The set of all 1-voxels in a given 3-D picture is called a (3-D) object. It is assumed that the first row, the I-th row, the first column and the J-th column of each plane, the first and K-th plane are all filled with 0-voxels.

[Definition 1] For any voxel $X = (i,j,k)$, three sets of voxels $N^{(6)}(X)$, $N^{(18)}(X)$, and $N^{(26)}(X)$ are defined as below and are called the 6-neighborhood (6-n.), 18-neighborhood (18-n.) and 26-neighborhood (26-n.) of X, respectively. (Fig. 2)

$$\begin{aligned} N^{(6)}(X) &= \{(p,q,r);\ |p-i| + |q-j| + |r-k| = 1\} \\ N^{(18)}(X) &= \{(p,q,r);\ 0 < |p-i|^2 + |q-j|^2 + |r-k|^2 \leq 2\} \qquad (1) \\ N^{(26)}(X) &= \{(p,q,r);\ \max(|p-i|, |q-j|, |r-k|) = 1\} \end{aligned}$$

Basing upon these definitions of the neighborhood, three types of connectivity (k-connectivity or k-c., where k=6, 18, or 26) are defined in the same way as those of 2-D pictures [1]. In order to keep theoretical consistency we should introduce the fourth type of connectivity called the 18'-connectivity, and should adopt a set of connectivity types for 1-voxels and 0-voxels as in Table 1. For details see Ref. [2]. By regarding above neighbors as of unit distance from the voxel X, three distance functions are obtained. They are denoted by d_6, d_{18}, and d_{26}, respectively. Consider a particular 1-voxel X in an object. Let $E^{(k)}(X)$ and $E^{(k)}(\bar{X})$ denote the Euler number (genus) of the object before and after changing the value of the 1-voxel X into 0 (i.e., deleting X), respectively. A superfix k represents the type of connectivity (k=6, 18, 18', and 26).

[Definition 2] The connectivity number of a 1-voxel X, $Nc^{(k)}(X)$ is defined as

$$Nc^{(k)}(X) = E^{(k)}(\bar{X}) - E^{(k)}(X) + 1 \qquad (2) \qquad \text{where } k=6,\ 18,\ 18',\ \text{or } 26.$$

The genus $E^{(k)}(X)$ is obtained by the triangulation method or the simplex counting method [2]. Formulas to calculate the connectivity number are explicitly given in [2] and [3] for all of four connectivity cases.

If changing a 1-voxel X to a 0-voxel causes no change in any of three kinds of numbers, numbers of connected components, holes and cavities, then it is said that the 1-voxel X is deletable (strictly speaking, k-deletable where k (= 6, 18, 18', 26) means the type of connectivity). Obviously, if a 1-voxel X is k-deletable, $Nc^{(k)}(X) = 1$. However, the converse is not always true. In order to guarantee invariance of topological structure, the following three features must be taken into consideration. (Fig. 3)

Table 1 Consistent pairs of connectivity types

n (for 1-voxel)	$\bar{n}$ (for o-voxel)
6	26
18	18'
18'	18
20	6

[Definition 3] Component index $R^{(k)}(X_{20})$ = number of k-connected components which are connected to X_{20} and exist in the 18-n. for k=6 and 18' (in the 26-n. for k=18 and 26).

Hole index $H^{(k)}(X_{20})$ = number of holes newly generated by deleting the 1-voxel X_{20} in the 3x3x3 local area of Fig. 3.

Cavity index $Y^{(k)}(X_{20})$ = number of cavities newly generated by deleting the 1-voxel X_{20} in the 3x3x3 local area of Fig. 3.

The superscript k denotes the types of connectivity. The set of three features (R(X), H(X), Y(X)) is called the connectivity index. Then the following theorem holds.

[Theorem 1] The necessary and sufficient condition that a 1-voxel X is k-deletable is that $R^{(k)}(X) = 1$ and $H^{(k)}(X) = Y^{(k)}(X) = 0$, or equivalently $Nc^{(k)}(X) = R^{(k)}(X) = 1$.

Derivation of Theorem 1 is shown in [2] with several experimental results. All possible values of the connectivity number and the connectivity index are shown in [2] and [3] with several examples.

3. Shrinking of a 3-D object

Shrinking is a procedure to reduce any 3-D simply connected object into an isolated point without changing topology (without causing split and merge of any connnected components, without generating any of holes and cavities, and without losing any of existing holes and cavities). A shrinking algorithm for 3-D objects is derived immediately as follows by converting 1-voxels into 0-voxels sequentially.

[Algorithm I] (Shrinking) Scan a given 3-D picture by the mode I in Fig. 4, check deletability at each 1-voxel, and change any deletable 1-voxel into the 0-voxel as soon as it is found. Repeat all these steps until all voxels are found to be undeletable.

Any simply connected object in the 3-D space is reduced to an isolated 1-voxel. If an object has a cavity of a hole, a thin figure preserving its cavity and hole is obtained. A procedure to check deletability of a 1-voxel is given in Ref. [2] and [3] with examples.

4. Thinning

A thinning algorithm is the one to transform a 3-D object into a figure with 1-voxel width almost everywhere, preserving the topological structure and preventing excessive degenration of the object. A sequential type of thinning algorithm is derived basing upon the same principle as that of the shrinking algorithm presented above. Before presenting the algorithm, let us clarify our objective more by defining a digitized 3-D surface figure.

[Definition 3] An object in a 3-D digitized picture is called a 3-D surface figure if and only if it satisfies at least one of the following (a) and (b).

(a) An object does not contain any of 3-D simplexes (Fig. 5).

(b) An object does not contain a deletable 1-voxel.

We consider that 3-D thinning is a procedure to transform any object in a given 3-D image into a 3-D surface figure and that the 3-D thinning procedure should satisfy all of the following requirements.

[Requirements] (1) Topological properties of the original object should be preserved. (2) A resulting figure should be the 3-D surface figure. (3) A resulting figure should be located along the center line (center surface) of an original object. (4) An original object should not degenrate excessively during a thinning procedure.

Let us present an algorithm which was derived by the authors and their colleagues [4].

[Algorithm II] (Thinning)

Input: $F = \{f(i,j,k)\}$: A 3-D binary picture where $f(i,j,k)$ denotes a density value at the voxel (i,j,k).

m: The type of connectivity (m = 6, 18, 18', or 26)

Output: $F = \{f(i,j,k)\}$: A 3-D ternary picture. A set of voxels with positive densities are regarded as an object. All objects have been changed into surface figures.

Voxel-preservation condition (VPC)

VPC 1: A voxel (i,j,k) is not contained in any of 3-D simplexes.

VPC 2: A voxel (i,j,k) is not deletable.

A voxel satisfying VPC 1 or VPC 2 is called a preserved-voxel.

```
[STEP 1]  Detection and grouping of border
          voxels
for all (i,j,k)'s do*
    if f(i,j,k) = 1 then
    begin
     if f(i,j,k+1) = 0 then f(i,j,k) ← 7;
     if f(i,j+1,k) = 0 then f(i,j,k) ← 6;
     if f(i+1,j,k) = 0 then f(i,j,k) ← 5;
     if f(i-1,j,k) = 0 then f(i,j,k) ← 4;
     if f(i,j-1,k) = 0 then f(i,j,k) ← 3;
     if f(i,j,k-1) = 0 then f(i,j,k) ← 2;
    end
    else no operation is performed
```

```
[STEP 2]  Subcycles
for bordertype ← 2 to 7 do
begin

[STEP 2.1]  Detection of preserved-voxels
for all (i,j,k)'s do
    begin
     if(f(i,j,k) = bordertype ∩ (VPC1 ∪ VPC2)
     then f(i,j,k) ← 10
     else no operation is performed
    end

[STEP 2.2] Elimination of deletable voxels
for all (i,j,k)'s do
    begin
     if f(i,j,k) = bordertype and
        (i,j,k) is deletable
     then f(i,j,k) ← 0
     else f(i,j,k) ← 10
    end
end

[STEP 3] Test of the terminating condition
if no point changed through [STEP 2]
then stop
else go to [STEP 1]
```

* By the statement "for all (i,j,k)'s do" we mean that every voxel is processed sequentially according to the mode I in Fig. 4.

The step 1 classifies every border voxel into six groups according to which side of the voxel a 0-voxel is located. Only one of these groups is processed in each of subcycles in the Step 2. This garantees that the third requirement is satisfied approximately and sometimes exactly. The first requirement is satisfied since only a deletable voxel is eliminated in the Step 2.2. The second requirement is obviously satisfied since the voxel preservation conditions is coincident with Def. 3. The test of the VPC 1 (detection of 3-D simplexes) is performed by pattern matching on the 2x2x2 local area. The VPC 2 (deletability) is tested by calculating the connectivity index using pseudo-Boolean expressions defined on the 3x3x3 local area. Thus, the above algorithm is implemented by an iterative local operation of the sequential type with the 3x3x3 neighborhood.

Another algorithm was proposed by Y.F. Tsao and K.S. Fu [5]. Characteristics of both methods are compared in Table 2.

Table 2 Comparison of 3-D thinning methods

method	KYT [1982] [4]	TF [1981] [5]
type	sequential	parallel
connectivity	6,18,18',26	6,26
deletability test	connectivity index	check plene
subcycle	6	6
topology	preserved	preserved
degeneration	a little less. mainly one layer along the object surface	a little more. mainly crossing part of hyper-planes
spurious branches	a littel more	a little less

5. Distance trasformation and skeleton

General theory of the distance transformation on a 2-D digitized picture was presented in Ref. [6]. The theory and algorithms shown there are extended to a 3-D digital image in a straightforward way. Here let us give a sequential algorithm of the 3-D distance transformation which is most suitable for conventional digital computer.

[Algorithm III] (3-D distance transformation-sequential type)

Input: $F = \{f_{ij}\}$, 3-D bianry picture

Output: $G = \{g_{ij}\}$, Distance picture, in which $g_{ij} = \begin{cases} \text{distance value, if } f_{ijk} = 1 \\ 0, \quad \text{if } f_{ijk} = 0. \end{cases}$

(1) Initialization

$$g_{ijk} = \begin{cases} M, \text{ if } f_{ijk} = 1 \\ 0, \text{ if } f_{ijk} = 0, \text{ for } \forall(i,j,k) \end{cases}$$

where M is a sufficiently large positive integer.

(2) Iteration

A picture $G^{(1)}$ and $G^{(2)}$ are calculated as follows.

(a) $g^{(1)}_{ijk} = \min \{g_{ijk}, \min_{(p,q,r)\in X} \{g^{(1)}_{pqr}\} + 1\}$ for $\forall(i,j,k)$

Computation of the value $g^{(1)}_{ijk}$ for each voxel is made according to the scanning mode I in Fig. 4.

(b) $g^{(2)}_{ijk} = \min \{g^{(1)}_{ijk}, \min_{(p,q,r)\in Y} \{g^{(2)}_{pqr}\} + 1\}$ for $\forall(i,j,k)$

The scanning mode II in Fig. 4 is adopted in this step.

Two sets of voxels X and Y should be selected as in Fig. 6, corresponding to the type of distances, d_6, d_{18}, and d_{26} used here. Physically the same memory area may be assigned to all of G, $G^{(1)}$ and $G^{(2)}$, if G and/or $G^{(1)}$ need not be preserved.

A parallel type algorithm of the distance transformation is presented in [6].

6. Border following

Border follwoing is a procedure to trace all border pixels systematically. It transforms a digitized 3-D picture into a table (list)of coordinates of all border voxels arranged according to the neighboring relation in the original picture. Border following algorithms have been extensively studied by G. Herman et al. [7]. Let us present here a different method for border following, which traces 1-voxels (border voxels) themselves only instead of tracing surfaces of 1-voxels such as the method in Ref. [7].

[Definition 4] A voxel (with the density value m (= 0 or 1)) of an n-connected component C is called a border voxel if and only if it has the density value m and has at least one voxel with the density value $\bar{m} = 1 - m$ in its $\bar{n}$-neighborhood.

[Definition 5] (surface neighbor) Given a 1-voxel P_1 and an integer M, a set of all 1-voxels P_2 satisfying the following conditions (i) and (ii) is called the (m-connected) surface neighbor of P_1 for M, and denoted by $NB^{(m)}(P_1, M)$.

(i) $P_2 \in N^m(P_1)$, if P_1 is a 1-voxel. $P_2 \in N^{\bar{m}}(P_1)$, if P_1 is a 0-voxel.

(ii) For voxels located relative to the voxel P_1 as shown in Fig. 7, it holds that

$$f_{P1} = f_{P2} \neq f_{q1} = f_{q2} \text{ and } (f_{q1} = f_{q2} = M \text{ for } M = 0, 1) \cap (Mf_{q1} = Mf_{q2} = M \text{ for } M \neq 0, \neq 1).$$

Using these concepts, a border following algorithm is described as follows.

[Algorithm IV] (Border following - voxel trace type)

Input: $F = \{f_{ij}\}$: binary picture m: type of connectivity

Output: OUTPUTLIST: List of coordinates of border voxels

Work area: TEMP : a queue, W: a voxel corresponding to the coordinates currently processed

```
(I) Initialization  MARK0 ← -1, MARK1 ← 1
(II) Starting voxel search
    Start scanning of an input picture F from the previous starting voxel (from the north-
west corner of the first plane, for the first search) according to the mode I raster scan.
If a starting voxel pair (S0, S1) is found, go to (III), where (S0, S1) is a pair of a 0-
voxel and a 1-voxel 6-adjacent each other in the same scan line. Denote by P the object
including S1, and the compliment of P by Q.
(III) Branch
    MARK 1 ← MARK1+1, MARK0 ← -MARK1
If the voxel S0 has a mark, go to V, otherwise go to IV.
(IV) Tracing of border voxels of the set Q. (Marking of 0-voxels)
    (1) Add the coordinates of S0 to the queue TEMP.
    (2) Mark of S0 ← MARK0
    (3) If TEMP has no element, then go to V, else go to (4)
    (4) Take one element out of TEMP and denote the corresponding voxel by W.
    (5) Extract all voxels in $NB^{(m)}(W, 1)$ having no mark in the input picture F.
        Give them the mark MARK0, and add their coordinates to TEMP.
```

(6) Go to (3).

(V) Tracing of the desired border voxels

(1) Set a segmentation code in the OUTPUTLIST.
(2) Add the coordinate of S1 to the queue TEMP and OUTPUTLIST, and give mark MARK1 to S1.
(3) If TEMP has no element, then go to II, else go to (4).
(4) Take the first element out of TEMP and denote the corresponding voxel by W.
(5) Extract all voxels in $NB^{(m)}$(W, mark of S0) which do not have the MARK1 in the input picture F. Give them the mark MARK1 and add their coordinates to TEMP and OUTPUTLIST.
(6) Go to (1).

This algorithm has several desirable properties as follows.
(1) All border voxels are extracted. They are grouped together corresponding to each surface.
(2) Every voxel contained in the same surface appears exactly once in the output list. If a particular voxel is included in n different surfaces, the voxel is passed n times.
(3) Border voxels extracted have a tree (or graph) structure, in which a node corresponds to each border voxel and a branch is drawn from a node A to a node B if A is in the neighborhood of B and B is visited just after A.
(4) The algorithm is applicable for any of 6-, 18-, 18'- and 26-connectivity.

The original object is exactly restored from the output list (tree of border voxels) and the coordinates of starting voxel pair. (For a reconstruction algorithm, see Ref. [8]). Both of coordinate representation and chain encoding are available for the output list.

Concluding remarks

Features of algorithms considered are compared with those for a 2-D picture in Table 3, through 7 including experimental results in Fig.8.

Acknowledgements

Authors wish to thank Prof. T. Fukumura for his encouragement. Experiments were performed by Mr. H. Suzuki of Toyohashi University of Technology and Mr. T. Matsumoto of Nagoya University. This work was supported in part by the Ministry of Education, Japanese Government, under Grant-in-Aid for Scientific Research (No. 57380004)

References

1. S.N. Srihari: "Representation of three-dimensional digital images", Computing Surveys, 13, 4, pp.399-424 (Dec. 1981)
2. T. Yonekura, S. Yokoi, J. Toriwaki and T. Fukumura: "Connectivity in a three-dimensional digitized picture", Tech. Res. Rep., No. 8209,Dep. of Inform. Science, School of Eng., Nagoya Univ., Nagoya, JAPAN (May 1982)
3. J. Toriwaki, S. Yokoi, T. Yonekura and T. Fukumura: "Topological properties and topology-preserving transformation of a three-dimensional binary pictures", Proc. 6th ICPR, PP.414-419 (Oct. 1982)
4. Y. Kawase, S. Yokoi, J. Toriwaki and T. Fukumura: "A thinning algorithm for three-dimensional binary images", Paper of Technical Group on Pattern Recognition and Learning, PRL82-69 (Jan. 1981) (In Japanese)
5. Y.F. Tsao and K.S. Fu: "A parallel thinning algorithm for 3-D pictures", Computer Graphics And Image Processing, 17, pp.315-331 (1981)
6. J. Toriwaki and S. Yokoi: "Distance transformations on a digitized picture and their applications", in A. Rosenfeld and L. Kanal (eds.) : Progress in Pattern Recognition Vol.1 pp.187-264 American Press, 1982
7. J.K. Udupa, S.N. Srihari and G.T. Herman: "Boundary detection in multidimensions", IEEE Trans. PAMI-4, 1, pp.41-50 (Jan. 1982)
8. T. Matsumoto, S. Yokoi, J. Toriwaki and T. Fukumura: "A border-following algorithm for three-dimensional binary images", Paper of Technical Group on Pattern Recognition and Learning, PRL83-10 (May 1983) (In Japanese)
9. N. Okabe, J. Toriwaki and T. Fukumura: "Path and distance functions on a three-dimensional digitized picture", Pattern Recognition Letters (In printing)
10. T. Sakabe, S. Yamamoto, J. Toriwaki and T. Fukumura: "Enumeration of local patterns in two- and three-dimensional digitized binary pictures", ibid (8), PRL80-85 (Feb. 1981)

Table 3 3D-2D comparison - basic concept

	3-D	2-D
connectivity/ neghborhood	6,18,18',26	4,8
local area	3x3x3, 2x2x2	3x3, 2x2
sampling grid (voxel)	cubic(cube), face-centered cubic (rhonbic dodecahedron)	square(square) hexagonal(hexagon)
topology	numbers of components, holes and cavities	numbers of holes and components
deletability	same as the above	Euler number
local pattern	285288 for 3x3x3 region[10]	51 for 3x3 region [10]
distance measure	three basic types, and four variations[1]	two basic types, few variations

Table 4 3D-2D comparison - thinning

	3-D	2-D
connectivity	6,18,18',26	4,8
local area	3 x 3 x 3	3 x 3
type	sequential parallel(6-c, 26-c)	sequential parallel
deletability	connectivity index, check plane	connectivity number, crossing number
subcycle	6	4, 2

Table 5 3D-2D comparison - distance transformation

	3-D	2-D
neighborhood	general	general
Local area	general	general
type- sequential parallel	6-,18-,26-neighbors general, variable	4-,8-neighbors, octagonal general, variable
reconstruction from skeletons	proved[6]	proved[6]

Table 6 3D-2D comparison - border following

	3-D	2-D
connectivity	6,18,18',26	4,8
local area	2 x 2 x 2	2x2/3x3
output	tree, graph	linear list
chain code	6 or 26 directions	4 or 8 directions
variations	voxel trace face trace	pixel trace edge trace vertex trace

Table 7 Experimental results - computation time in sec.

		Thinning[1]	Shrinking	Border following[2]	Edge detection filtering*[3]
3D	6-c.	34.4 73.7	11.9	2.00 1.30	1.02
	18-c.	37.3 25.0	7.31	2.00 1.20	8.72 10.3
	26-c.	36.0	5.97	1.70	23.2 18.9
2D	4-c.		4.07	1.18 1.13	3.37 4.96
	8-c.	3.65	3.23	1.18 1.07	4.97

Input : Brain CT image (binarized) 128x128x20 voxels
including 138011-voxels
(*Brain CT image (grey) 128x128x35 voxels)
For 2D cases, 20(35) images of 128x128 pixels.
Methods : 1 KYT (left) and TF (right)
2 Voxel trace (left) and voxel-face trace (right)
Pixel trace (left) and vertex trace (right)
3 Prewitt type (left) and Laplacian (right)
Computer : FACOM M-200. Coded by FORTRAN.

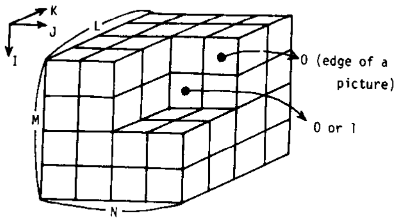


Fig. 1 Three-dimensional digitized picture

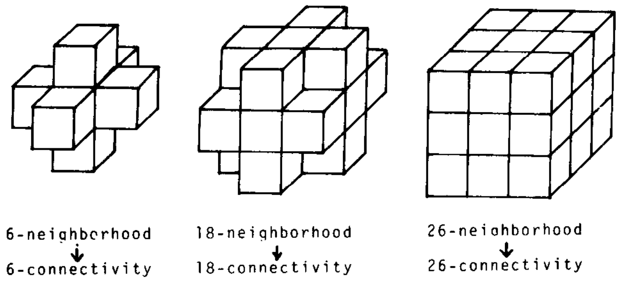

6-neighborhood ↓ 6-connectivity
18-neighborhood ↓ 18-connectivity
26-neighborhood ↓ 26-connectivity

Fig. 2 Neighborhood in the digitized 3-D space

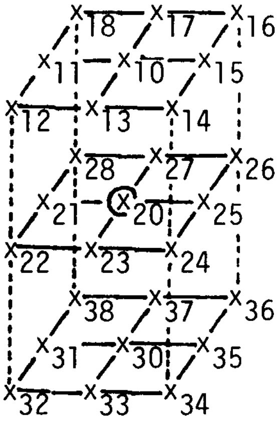


Fig. 3 26-neighborhood

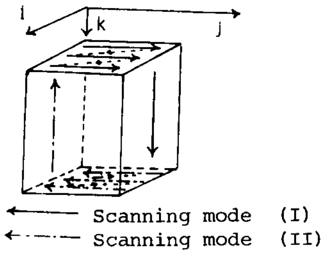
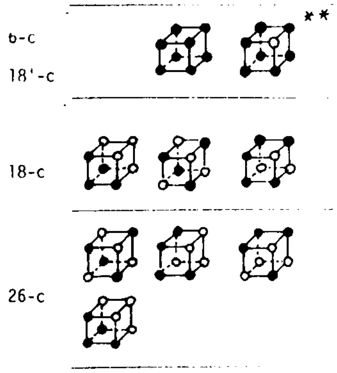


Fig. 4 Scanning mode



** Only for the 18'-c. cases

● : 1-voxel, O: 0-voxel

Fig. 5 All possible 3-D simplexes

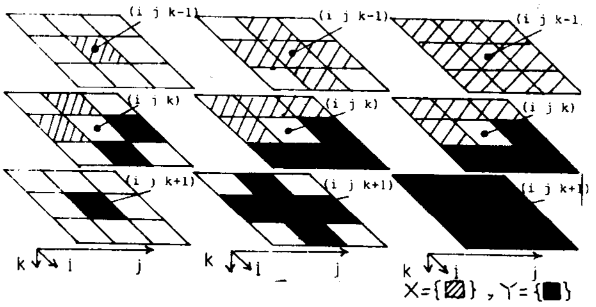
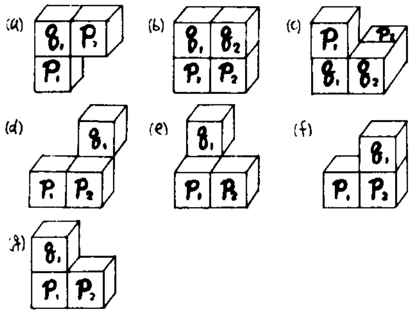


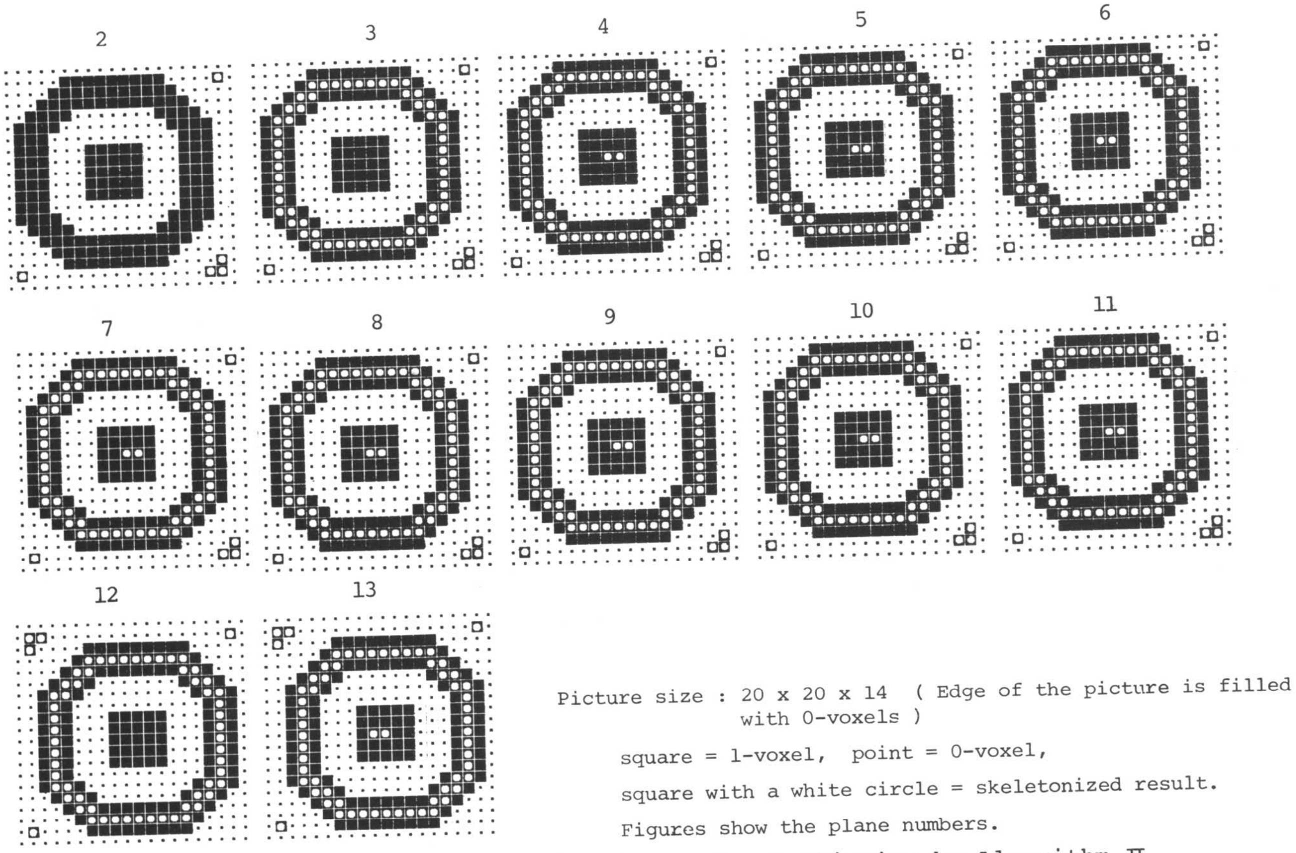
Fig. 6 Illustration for Algorithm III



d,e,f,g	for m = 6
a,c	for m = 18
e,f,g	for m = 18'
a,b,c	for m = 26

All patterns symmetric with respect to coordinate axes and coordinate planes are also considered.

Fig. 7 Illustration for Algorithm IV



Picture size : 20 x 20 x 14 (Edge of the picture is filled with 0-voxels)

square = 1-voxel, point = 0-voxel,

square with a white circle = skeletonized result.

Figures show the plane numbers.

Fig. 8 Example of thinning by Algorithm II

A new family of nonlinear edge detectors for noisy images

Nobuyuki Otsu and Tony Kasvand

Electrotechnical Laboratory, Niihari-gun, Ibaraki, Japan,
and
National Research Council, Ottawa, Ont., K1A 0R6, Canada.

Abstract

Many methods have been proposed to enhance and detect edges in gray-level images. Most of these are based on spatial differentiation to enchance gray-level changes, and have the common problem of being very sensitive to noise. In order to surpress the noise, some spatial averaging has been combined with the differentiation. This, however, tends to complicate the definition of the operator and also results in not very sharp edges.

In this paper, we consider the characteristics of edges and the statistical properties of noise in gray-level histograms taken from local regions of an image. The noise can then be regarded as a blurring process on the gray-level histograms. Therefore, the problem of edge detection for noisy images can be reduced to the problem of invariant feature extraction under one-dimensional blurring. By applying a theory of blurring-invariant feature extraction, a new family of nonlinear edge detectors is derived. The resultant operators are simple, based on central moments of the gray levels within a local window, and definable independently of edge orientation in the window. The operators are quite insensitive to noise, and their effectiveness has been confirmed by experiments.

Introduction

Edge extraction is one of the most essential techniques in image processing and computer vision, for example, for segmenting an image into different regions. Contours are intuitive features to ourselves. Generally, edges are defined as boundaries of regions where we can observe a sharp and remarkable change of some characteristic, for example gray level, colour, or texture. Therefore, edge extraction is performed by enhancing the change of such characteristics. Edge detection is performed by scanning an entire image with a local operation window. However, there are also edges which we may see in an image due to our understanding of the scene and recognition of the objects within it. Such edges may be too subtle for any local operator to detect.

Many edge extractors have been proposed [1-4]. Most of those are based on spatial differentiation, and the variety is mainly due to the diversity of definitions of the differentiation operation for digitized images. First-order derivatives are frequently used for extracting edges in a specified direction. Gradient operators are used for direction-free edge extraction [1]. Second-order derivatives (Laplacian type operators) are also commonly used, since they are sensitive not only to edges but also to corner points and isolated points. Edge operators based on differentiations are generally sensitive to noise, therefore some averaging operation to smooth out noise needs to be combined in practical applications to noisy images. However, extension of local window size for smoothing tends to complicate the definition of differentiation and also results in not very sharp edges. In order to obtain sharper and thinner edges, an iterative use of a Laplacian type operator has been proposed [2].

On the other hand, several methods are based on least square model fitting. Heuckel proposed a method of estimating the parameters of a step edge model in a circular local region [3]. However, the method is complicated and requires considerable computation. In the regression analysis type of approach [4], a regression plane ax+by+c (or quadratic plane) is fitted within the local window in the least square sense. The coefficients a and b are utilized as smoothed x and y derivatives, respectively, and the square root of a^2+b^2 or /a/+/b/ can be regarded as a smoothed gradient. The least square error might be considered as a smoothed version of Laplacian. This approach is interesting in the sense of automatically generating some kinds of smoothed differential edge operators in a unified manner. However the window size problem still remains, namely; the larger the local window to smooth out noise the less sharp the edges will be. Therefore, the essential problem of edge extraction is how to enhance edges (signals) while reducing noise, i.e., how to construct edge operators which achieve high signal-to-noise (S/N) ratio.

In this paper, we shall propose a new family of nonlinear edge extractors, formulating the problem of edge extraction for noisy images as a problem of feature extraction under a one-dimensional blurring process on the histograms of a chracterizing measure.

Blurring-invariant feature extraction

Noise and blurring on histograms.

We shall consider classification of objects in general. A basic approach would be to measure any characteristic of the objects and to observe its histogram. If the objects consist of two different classes and the characteristic is capable of discriminating the two classes, then the histogram will show a bimodal distribution. As an extreme, if the two classes are both completely uniform, the histogram will show two separate sharp peaks (spikes). On the other hand, if the objects of each class have variance, the variations will reflect themselves as a scatter around each peak. Such a scatter on the histogram will approach the normal distribution if the original variations of the objects are due to many superimposed effects caused by various probabilistic factors. Namely, the ideal histogram of two classes, as represented by two distinct and separate delta-functions, starts to resemble a distribution resulting from the convolution of the delta-functions with a normal distribution function (blurring process). The measurement of the characteristic itself is generally accompanied by error (noise) and results in similar blurring of the histogram. Consequently, the one-dimensional Gaussian blurring process on a histogram may be regarded as an essential process encountered in object classification in noisy circumstances.

In the case of the edge extraction problem in image proccessing, the histogram of any characteristic of the pixels in a local region (observation window), for instance the gray-level histogram, will show two distinct sharp peaks when an ideal step edge is located there. The histogram will show one unique sharp peak when the local region is flat (no edge). The typical histograms, however, in practice show blurred shapes due to the presence of additive noise (See Figure 1).

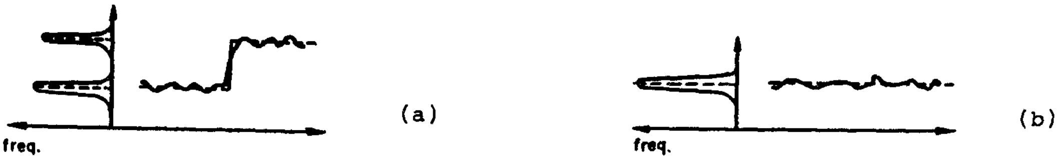


Figure 1. Examples of histograms for a step edge (a) and a flat region (b).

Therefore, the problem of stable edge extraction in noisy circumstances will be reduced to the problem of one-dimensional blurring invariant feature extraction from the histograms, or how to extract features which are invariant to the blurring of histograms and which are capable of discriminating the differences of such histogram patterns.

Blurring invariant histogram-moments.

Let x be any characteristic measured on the objects, and f(x) be its distribution function (histogram). Then the blurring transformation T(b) on f(x) is given by the following convolution with a normal distribution function treated as a point spread function.

$$T(b)f(x) = 1/\sqrt{4\pi b}\,\exp(-x^2/4b)*f(x) \qquad (1)$$

According to the theoretical results on blurring invariance [5], it has been shown that linear invariant features of one-dimensional distribution patterns under the Gaussian blurring transformation are limited only to integrals and centers of gravity, i.e. 0th- and 1st-order moments. In the case of gray-level histograms, those respectively correspond to the number of pixels and the average gray-level within a local window, and are of no use for edge extraction. Therefore, we must consider nonlinear invariant features. In the present paper, we shall consider moment-invariants by applying the construction theory of nonlinear invariants [6] to the moments, because the moments are simple to compute and are also basic as admissible features of various invariant transformations. The moments of pth-order are defined by

$$m_p = \int x^p f(x)dx = \langle x^p,\ f(x)\rangle \qquad (2)$$

where, for simplicity, $\langle g(x),\ f(x)\rangle$ hereafter symbolically represents an integration with respect to x. The blurring transformation is self-conjugate, and its generator (infinitesimal transformation) and its conjugate, t and t* in this one-dimensional case, are given by the second-order differential operator d^2/dx^2. The induced generator $\hat{t}$ on m_p is, therefore, obtained as follows.

$$\hat{t}m_p = \langle x^p,\ tf(x)\rangle = \langle t^*x^p, f(x)\rangle = \langle d^2/dx^2 x^p, f(x)\rangle = p(p-1)m_{p-2} \qquad (3)$$

Let any nonlinear quantity (feature) be represented by a scalar function of moments $h(m_0, m_1, \ldots, m_M)$. Then, by applying the theory [6], a necessary and sufficient condition for h to be invariant under the blurring transformation is given by the following partial differential equation.

$$\sum_{i=0}^{M} \frac{\partial h}{\partial m_i} \hat{t}m_i = 0 \qquad (4)$$

This can be reduced to the following equivalent subsidiary differential equation system.

$$\frac{dm_0}{0} = \frac{dm_1}{0} = \frac{dm_2}{2m_0} = \frac{dm_3}{6m_1} = \ldots\ldots\ldots = \frac{dm_M}{M(M-1)m_{M-2}} \; (= db) \qquad (4')$$

The elementary invariants are given by M independent solutions of this equation. The first and second terms yield $dm_0=0$ and $dm_1=0$, which confirms the previous statement that the 0th- and 1st-order moments are invariant to blurring. The third and forth terms show that the second and third moments change additively like $\tilde{m}_2 = m_2 + 2m_0b$ and $\tilde{m}_3 = m_3 + 6m_1 b$, respectively by the blurring transformation, where the symbol ~ denotes the value transformed by the blurring transformation. Elementary invariants are, for example, obtained from the third and forth terms as $6m_1dm_2=2m_0dm_3$ therefore $I_3=m_0m_3-3m_1m_2$, and also from the third and fifth terms as $12m_2dm_2=2m_0dm_4$, therefore, $I_4=m_0m_4-3m_2^2$, and so forth. The above discussion is also valid for the normalized central moments, which are invariant to translation (uniform shift) of x and amplitude change of f(x),

$$\mu_p = \left\langle (x-m_1/m_0)^p , \; f(x)/m_0 \right\rangle \qquad (5)$$

because m_0 and m_1 are invariant under the blurring transformation. Of course, we have the identities $\mu_0=1$ and $\mu_1=0$ in this case. Consequently, for the central moments we obtain the following blurring invariants.

$$I3 = \mu_3 \; , \quad I4 = \mu_4 - 3\mu_2^{\,2}, \; \ldots. \qquad (6)$$

The second-order central moment, i.e. variance, is shown to change its value additively as

$$\tilde{\mu}_2 = \mu_2 + 2b \qquad (7)$$

where $2b=\sigma^2$ is just the same in value as the variance of noise. This means that we might use the 2nd-order central moment as a quasi-invariant in edge extraction, since it only increases in value uniformly by increased blurring. The one-dimensional blurring-invariant moment features obtained above will be widely applicable as blurring-invariant nonlinear features not only to edge extraction in noisy images but also to various recognition purposes for one-dimentional patterns which are under blurring transformation.

Application to edge extraction in noisy images

We shall apply the blurring-invariant histogram-moments (hereafter denoted simply by BIHM) in general to edge extraction from noisy images. Let the size of the local window (mask) scanning over an entire digital image (size N by N) be M by M, for instance M=2L+1 (L=1,2,3,...). Denote the characteristic values of the pixels within the local mask, for example the gray-levels, by $x_i \in [0,1,\ldots,K]$ $(i=1,2,\ldots,M^2)$. Then BIHM are simply given by the following terms.

$$m_p = \sum_{i=1}^{M^2} x_i^{\,p} \quad (\, m_0 = M^2) \qquad (8)$$

$$\mu_p = \sum_{i=1}^{M^2} (x_i - m_1/m_0)^p / M^2 \quad (\, \mu_0 = 1, \;\; \mu_1 = 0 \,) \qquad (9)$$

It should be mentioned that these are moments of the values within the local mask and can be defined quite independently of the mask size and also of the direction of the edge located there. It should also be noticed that such definitions are straightforward also for one-dimensional cases, such as edge detection in gray-level image profiles or signals in time, where the patterns and masks are given as 1 by N and 1 by M, respectively.

The histogram of the characteristric values within the local mask will show two distinct sharp peaks when the mask is located on the ideal step-edge and will show one unique sharp peak for ideal flat regions. Therefore, if such a difference can be discriminated in terms of BIHM, we can construct edge detectors which are expected to be quite insensitive to noise in images. Consequently, the problem is how to represent the "edge value" in terms of BIHM, or how to construct a measure which takes on a larger value for more distant two peaks and takes a lesser value for two closer peaks in the histogram (desirably zero for one unique peak).