General Disclaimer

One or more of the Following Statements may affect this Document

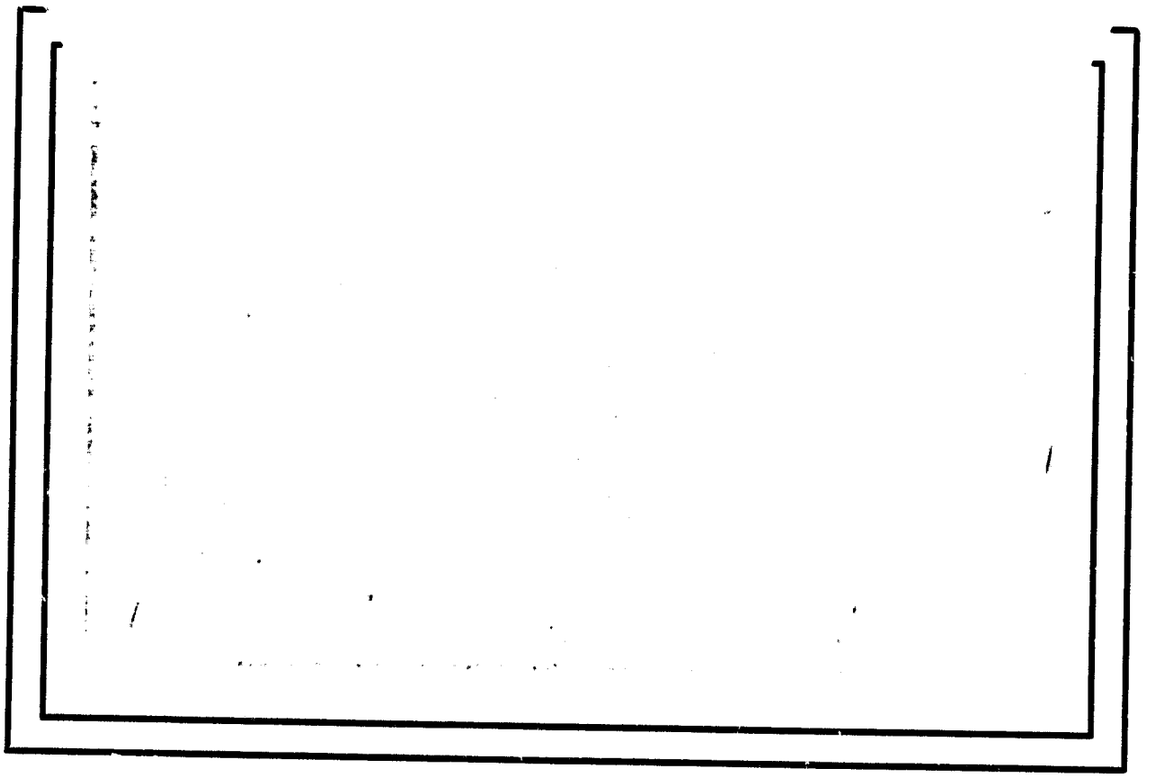
- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.



UNIVERSITY OF MARYLAND
COMPUTER SCIENCE CENTER
COLLEGE PARK, MARYLAND

Technical Report TR-156 May, 1971
AT-(40-1)-3662

CONNECTIVITY AND GENUS IN THREE DIMENSIONS

Chan Mo Park
and
Azriel Rosenfeld

Algorithms are presented for labeling connected objects in a binary three-dimensional array, for counting such objects, and for computing the genus of the array.

The support of the U.S. Atomic Energy Commission, under Contract AT-(40-1)-3662, is gratefully acknowledged. The help of Mrs. Eleanor B. Waters in preparing the manuscript is also deeply appreciated.

1. Introduction

A three dimensional volume of "digital space" may be represented by a stack of planes each of which contains a cross-section of the volume. Objects in such a volume can be specified by assigning the value 1 to their points, while points in the space between or within the objects have value 0.

This paper describes an algorithm for assigning labels to connected objects, so that each point of a given object receives the same label, while points belonging to different objects receive different labels. In particular, this algorithm can be used to count the objects (number of objects = number of distinct labels used) and to compute their volumes (volume of an object = number of times its label occurs).

The principle employed in the construction of the algorithm is to scan the volume element by element in a certain sequence and to examine each point found to have the value 1. For each such point which is not connected to any object found previously, a new label is assigned, the number of objects is incremented by 1 and the volume for the new object is initiated to be 1. If the point is connected to just one object found earlier, the volume of the object to which this element is connected is incremented

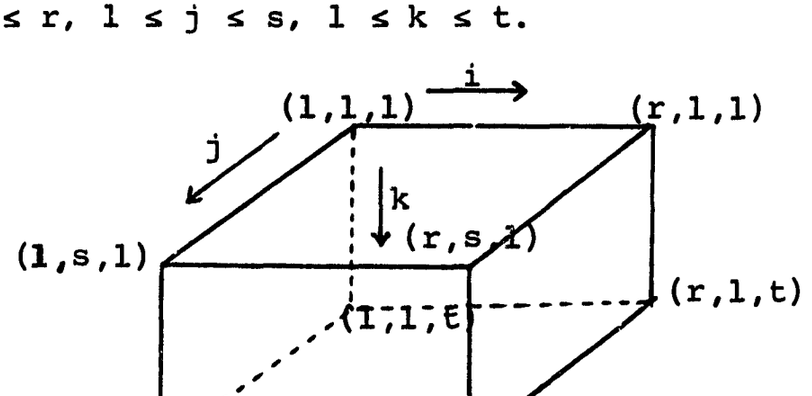
by 1. However, the point may be connected to several objects found earlier but not previously known to be connected. In this case, the volumes of these objects must be combined and the number of objects must be reduced. Also included in this paper is an algorithm to compute the three-dimensional genus.

A detailed discussion of the topological properties of three-dimensional arrays may be found in S. B. Gray, "Local properties of binary images in two and three dimensions", Information International, Inc., January 21, 1970.

2. Definitions and notation

An explanation of the definitions and notation used in the following sections is given here.

a) The volume to be analyzed is an r by s by t hexahedron (rectangular parallelepiped). An arbitrary point (or element) in the volume will be denoted by $P_{i,j,k}$, where $1 \leq i \leq r$, $1 \leq j \leq s$, $1 \leq k \leq t$.



The volume can be regarded as a stack of t planes each of which consists of r by s points.

b) In defining connectivity, two elements are regarded as adjacent if the distance between them is one unit. Thus, any element is adjacent to six other elements: one element on the plane above, four elements on the same plane, and one element on the plane below.

c) The volume is processed in the following sequential order:

i increases most rapidly
j increases next most rapidly
k increases least rapidly

Thus the order of scan is:

$P_{1,1,1}\ P_{2,1,1}\ \cdots\ P_{r,1,1}$

$P_{1,2,1}\ P_{2,2,1}\ \cdots\ P_{r,2,1}$

$\vdots$

$P_{1,s,1}\ P_{2,s,1}\ \cdots\ P_{r,s,1}$

$P_{1,1,2}\ P_{2,1,2}\ \cdots\ P_{r,1,2}$

$P_{1,2,2}\ P_{2,2,2}\ \cdots\ P_{r,2,2}$

$\vdots$

d) The following are the major variables in the algorithm:

count = the number of objects in the volume

$label_{\lambda}$ = the label of the λth object found

$volume_{\lambda}$ = the volume of the λth object

$P_{i,j,k}$ = an arbitrary element

$L_{i,j,k}$ = the label of the element $P_{i,j,k}$

$neighbor_{h}$ = the index of the label of an element which is connected to the element being examined.

$oldlabel_{\alpha}$ = the original label which was assigned when the object was found

$newlabel_{\alpha}$ = the new label assigned to the object because it is found to be connected to an object labeled earlier.

λ = index for the arrays of labels and volumes

α = index for the arrays of oldlabels and newlabels

3. The labeling algorithm

a) Initialization

The number of objects, various arrays and the indexes are cleared:

$count \leftarrow 0$

$label_i \leftarrow 0$

$volume_i \leftarrow 0$

$oldlabel_i \leftarrow 0$

$newlabel_i \leftarrow 0$

$\lambda \leftarrow 0$

$\alpha \leftarrow 0$

b) Pick the next element, $P_{i,j,k}$.

If it is 0 go to (g).

If it is 1 go to (c).

c) If $k > 1$, examine the label $L_{i,j,k-1}$ of the neighbor of $P_{i,j,k}$ on the plane above.

d) Examine the labels of the elements in the same plane which could be connected to $P_{i,j,k}$ and which have already been processed. In general these are $L_{i,j-1,k}$ and $L_{i-1,j,k}$ (unless $P_{i,j,k}$ is on a face of the volume).

e) If at least one of these labels is not 0 go to (f). Otherwise, a new object has been found; set

$count \leftarrow count + 1$

$\lambda \leftarrow \lambda + 1$

$label_\lambda \leftarrow f(\lambda)$ where f is a label generation function*

$volume_\lambda \leftarrow 1$

$L_{i,j,k} \leftarrow label_\lambda$

and go to (g).

f) If $P_{i,j,k}$ has neighbors that have already been labeled,

i) Form an array "neighbor" of the indices (λ's) of the labels of these neighbors. Assume that there are ν such indices. Find the minimum element in the array, i.e., $\lambda_{min} = \min_{1 \le h \le \nu} (neighbor_h)$

ii) Adjust "count"

$count \leftarrow count - (\nu - 1)$

iii) Determine the label of the current point, and save it

$L_{i,j,k} \leftarrow label\ (\lambda_{min})$

iv) Update the volumes

$Volume_{\lambda_{min}} = \sum_{h=1}^{\nu} volume_{neighbor_h} + 1$

$Volume_{neighbor_h} = 0$ except for $neighbor_h = \lambda_{min}$

v) Change the L's which are connected to $P_{i,j,k}$ into $L_{i,j,k}$ ($\equiv label\ (\lambda_{min})$), because they may be connected to the next element to be scanned.

---

*E.g., for a UNIVAC 1108

$f(\lambda) = \text{"blank"} + \lambda/26 * 64 + mod(\lambda, 26)$

assuming that labels are single letters or pairs of letters, i.e., A,B,···Z, AA,AB,···ZZ.

vi) Enter the old and new labels into the arrays oldlabel and newlabel, respectively. This information will be used later to assign final correct labels to the points of objects.

$\alpha \leftarrow \alpha+1$

$\text{oldlabel}_{\alpha} = \text{label}\ (\text{neighbor}_h)$

$\text{newlabel}_{\alpha} = \text{label}\ (\lambda_{min})$

Repeat (vi) for $h = 1, 2, \ldots, \nu$ except for $\text{neighbor}_h = \lambda_{min}$.

g) If all the elements have been considered, go to (h); otherwise go to (b).

h) Update the array newlabel:

If

$$\text{newlabel}_i = \text{oldlabel}_j,$$

then

$$\text{newlabel}_i \leftarrow \text{newlabel}_j$$

Repeat this step until there is no further change.

i) Update the labels of objects

If

$$L_{i,j,k} = \text{oldlabel}_m,$$

then

$$L_{i,j,k} \leftarrow \text{newlabel}_m.$$

j) Stop

An example is given in Section 4 which shows successive stages in the application of the algorithm.

If the original volume is of large size, it may be necessary to use tapes to supply the input data and to record the labels assigned to the objects. If the size of the volume is such that the labels of all the points in one plane as well as those of the points in two rows in the next plane (i.e., rs + 2r points) can be stored in the core memory at the same time, only one tape (besides the input tape) is needed to record the labels of the objects. Otherwise, it is necessary to use two tapes. However, if the original data on the input tape can be destroyed, the input tape can be used to record the labels. In such cases no additional tape is necessary if the core memory can hold the labels of the points in two planes (i.e., 2rs points) at the same time.

4. An example of the labeling algorithm

The figure at the end of this section shows an input volume and the final labels assigned to the objects. When a point changes from one label to another during the process the old label is indicated inside parentheses beside the new label assigned to that point. The table below indicates the status after each plane is processed.

Plane	Count	λ	Label	Volume	Oldlabel	Newlabel	Remarks
k=1 (top plane)	3	1	A	20	0	0	3 different objects, A, B and C are found.
		2	B	2			
		3	C	2			
k=2	5	1	A	40	0	0	2 more objects, D and E are found.
		2	B	5			
		3	C	2			
		4	D	4			
		5	E	2			
k=3	6	1	A	60	0	0	A new object, F ($P_{10,3,3}$ and $P_{10,4,3}$) is found.
		2	B	10			
		3	C	2			
		4	D	8			
		5	E	4			
		6	F	2			
k=4	6 (6→5→6)	1	A	80	F	E	It is found that $P_{10,4,3}$(F) is connected to $P_{10,4,4}$ which is connected to $P_{10,6,3}$(E). Thus $L_{10,4,3}$ is changed to E and V_6 is added to V_5 before it is cleared. The count is reduced by 1. A new object (G) is found, which increases the count back to 6.
		2	B	10			
		3	C	2			
		4	D	8			
		5	E	9			
		6	F	0			
		7	G	3			

Plane	Count	λ	Label	Volume	Oldlabel	Newlabel	Remarks
k=5	6	1	A	120	F	E	When $P_{1,4,5}$ is processed, it is not connected to any points previously labeled. So H is assigned and the count is increased to 7. But H is found later to be connected to the object A via $P_{2,4,5}$, which reduces the count back to 6 and changes the label to A.
	(6→7→6)	2	B	10	H	A	
		3	C	2			
		4	D	8			
		5	E	9			
		6	F	0			
		7	G	6			
		8	H	0			

		Input Picture										Final Label											
	i	1	2	3	4	5	6	7	8	9	10												
Top plane k=1	j 1	0	0	1	1	1	1	1	1	0	0			A	A	A	A	A	A				
	2	0	0	1	0	0	0	0	1	0	0			A					A				
	3	1	0	1	0	0	0	0	1	0	1		B	A					A				C
	4	1	0	1	0	0	0	0	1	0	1		B	A					A				C
	5	0	0	1	0	0	0	0	1	0	0			A					A				
	6	0	0	1	1	1	1	1	1	0	0			A	A	A	A	A	A				
	7	0	0	0	0	0	0	0	0	0	0												
	8	0	0	0	0	0	0	0	0	0	0												
k=2		0	0	1	1	1	1	1	1	0	0			A	A	A	A	A	A				
		0	0	1	0	0	0	0	1	0	0			A					A				
		1	0	1	0	1	1	0	1	0	0		B	A		D	D		A				
		1	0	1	0	1	1	0	1	0	0		B	A		D	D		A				
		1	0	1	0	0	0	0	1	0	0		B	A					A				
		0	0	1	1	1	1	1	1	0	1			A	A	A	A	A	A				E
		0	0	0	0	0	0	0	0	0	1												E
		0	0	0	0	0	0	0	0	0	0												
k=3		0	0	1	1	1	1	1	1	0	0			A	A	A	A	A	A				
		1	0	1	0	0	0	0	1	0	0		B	A					A				
		1	0	1	0	1	1	0	1	0	1		B	A		D	D		A				E(F)
		1	0	1	0	1	1	0	1	0	1		B	A		D	D		A				E(F)
		1	0	1	0	0	0	0	1	0	0		B	A					A				
		1	0	1	1	1	1	1	1	0	1		B	A	A	A	A	A	A				E
		0	0	0	0	0	0	0	0	0	1												E
		0	0	0	0	0	0	0	0	0	0												
k=4		0	0	1	1	1	1	1	1	0	0			A	A	A	A	A	A				
		0	0	1	0	0	0	0	1	0	0			A					A				
		0	0	1	0	0	0	0	1	0	0			A					A				
		0	0	1	0	0	0	0	1	0	1			A					A				E
		0	0	1	0	0	0	0	1	0	1			A					A				E
		0	0	1	1	1	1	1	1	0	1			A	A	A	A	A	A				E
		0	0	0	0	0	0	0	0	0	0												
		0	0	0	0	1	1	1	0	0	0					G	G	G					
k=5		0	0	1	1	1	1	1	1	0	0			A	A	A	A	A	A				
		0	0	1	1	1	1	1	1	0	0			A	A	A	A	A	A				
		0	0	1	1	1	1	1	1	0	0			A	A	A	A	A	A				
		1	1	1	1	1	1	1	1	0	0		(H)	A	A	A	A	A	A	A			
		1	1	1	1	1	1	1	1	0	0			A	A	A	A	A	A	A			
		0	0	1	1	1	1	1	1	0	0			A	A	A	A	A	A				
		0	0	0	0	0	0	0	0	0	0												
		0	0	0	0	1	1	1	0	0	0					G	G	G					

5. Connectivity, volumes and genus

When the labels are not desired but only the number of objects (i.e., the connectivity) as well as their volumes, the labeling algorithm described above can be used with some modification. Although the labels are not required, it is necessary to assign temporary identifiers to the points in order to distinguish connected objects from one another. However, it is no longer necessary to keep the identifiers of all the points which have already been processed, but only those of the points on the plane above and the current plane. The other stages which are omitted in this case are those steps concerned with the arrays oldlabel and newlabel and with updating the labels of objects, i.e., steps (f vi), (h) and (i).

An even simpler algorithm can be used if one wants to compute the three-dimensional genus or Euler number. This is defined as O-H+C, where O is the number of connected objects, H is the number of holes in objects (topologically similar to the hole in a doughnut), and C is the number of cavities in objects (i.e., connected components of 0's, completely surrounded by 1's). It should be pointed out that in one and two dimensions the genus is defined as $E^{(1)} = O$ and $E^{(2)} = O-H$, respectively.

The genus can be computed by performing a local counting operation over the given volume. Let

C1 be the number of 1's

C2 be the number of 1 by 1 by 2 patterns of 1's (in all orientations)

C4 be the number of 1 by 2 by 2 patterns of 1's (in all orientations)

C8 be the number of 2 by 2 by 2 patterns of 1's

Then $E^{(1)} = C1-C2$

$E^{(2)} = C1-C2+C4$

$E^{(3)} = C1-C2+C4-C8$

A proof of this assertion, for $E^{(3)}$, is given in Section 6. In Section 7 an algorithm for computing $E^{(3)}$ is given.

6 Proof that $E^{(3)}$ = C1-C2+C4-C8

The proof is by induction on the number of 1's in the given volume. Specifically, it is shown that if the rightmost, uppermost 1 in the topmost plane is deleted, both $E^{(3)}$ = O-H+C and C1-C2+C4-C8 change by the same amount. (For the case where there are no 1's it is certainly true that $E^{(3)}$ = O-H+C = 0 and C1-C2+C4-C8 = 0.) Let P be this 1; then the neighbors of P whose values affect the Euler number are

```
a P   on the top plane
c d

e f   on the next plane below
g h
```

Case A. All three neighbors of P are 1's (i.e., a=d=f=1)

	Conditions	Changes in C1, C2, C4 & C8				Change in $E^{(3)}$	Remarks
		C1	C2	C4	C8		
i)	All other points are 1's	-1	-3	-3	-1	0	
ii)	All other points are 1's except g=0	-1	-3	-3	0	-1	A hole is created
iii)	c=0, e=h=1, g=0 or 1	-1	-3	-2	0	0	
iv)	c=0, {e=0, h=1 / e=1, h=0, g=0 or 1	-1	-3	-1	0	1	{a isolated / d isolated; No. of objects increased by 1
v)	c=e=h=0, g=0 or 1	-1	-3	0	0	2	a, d and f are separated. No. of objects increased by 2
vi)	c=1, {e=0, h=1 / e=1, h=0, g=0 or 1	-1	-3	-2	0	0	
vii)	c=1, e=h=0, g=0 or 1	-1	-3	-1	0	1	f isolated No. of objects increased by 1

Case B. Two neighbors are 1's

There are three such situations:

a=d=1 f=0

a=f=1 d=0

d=f=1 a=0

Since they are all similar, only the case a=f=1, d=0 will be considered

	Conditions	Changes in C1, C2, C4 & C8				Change in $E^{(3)}$	Remarks
		C1	C2	C4	C8		
i)	All other points are 1's except g=0 or 1	-1	-2	-1	0	0	
ii)	e=0, h=1, c and g=0 or 1	-1	-2	0	0	1	a and f are separated. No. of objects increased by 1
iii)	e=1, h=0, c and g=0 or 1	-1	-2	-1	0	0	
iv)	e=h=0, c and g=0 or 1	-1	-2	0	0	1	a and f are separated. No. of objects increased by 1

Case C. One neighbor is 1.

There are three such situations:

a=1 d=f=0

d=1 a=f=0

f=1 a=d=0

Only the case d=1, a=f=0 will be considered.

Conditions	Changes in C1, C2, C4 & C8				Change in E(3)
	C1	C2	C4	C8	
All other points are arbitrary	-1	-1	0	0	0

Case D. No neighbor is 1, i.e., a=d=f=0

Conditions	Changes in C1, C2, C4 & C8				Change in E(3)	Remarks
	C1	C2	C4	C8		
All other points are arbitrary	-1	0	0	0	-1	P was an isolated point. No. of objects decreased by 1

7. Algorithm for computing the genus

a) Initialization

C1 ← 0

C2 ← 0

C4 ← 0

C8 ← 0

b) Pick the next element, $P_{i,j,k}$. If it is 0, go to (d). Otherwise go to (c).

c) i) C1 ← C1+1

ii) Examine the elements in the (k-1)st and kth planes which are neighbors of $P_{i,j,k}$, or neighbors of those of its neighbors that have already been considered. In general, there are four such elements in the (k-1)st plane and three in the kth plane (provided $P_{i,j,k}$ is not on a face):

(k-1)st plane:	$P_{i-1,j-1,k-1}$,	$P_{i,j-1,k-1}$
	$P_{i-1,j,k-1}$,	$P_{i,j,k-1}$
kth plane:	$P_{i-1,j-1,k}$,	$P_{i,j-1,k}$
	$P_{i-1,j,k}$	

iii) Increment C2, C4 and C8 as follows:

$$C2 \leftarrow C2 + [P_{i-1,j,k} + P_{i,j-1,k} + P_{i,j,k-1}]$$

$$C4 \leftarrow C4 + [P_{i-1,j,k} \wedge P_{i-1,j-1,k} \wedge P_{i,j-1,k}] + [P_{i-1,j,k} \wedge P_{i-1,j,k-1} \wedge P_{i,j,k-1}] + [P_{i,j-1,k} \wedge P_{i,j,k-1} \wedge P_{i,j-1,k-1}]$$

$C8 \leftarrow C8+1$ provided (new C4)-(old C4) = 3 and $P_{i-1,j-1,k-1} = 1$

d) If all the elements have been considered go to (e). Otherwise go to (b).

e) $E^{(3)}$ = C1-C2+C4-C8; stop.