4de th

SUID-AFRIKAANSE REKENAARSIMPOSIUM
SOUTH AFRICAN COMPUTER SYMPOSIUM

HOLIDAY INN PRETORIA
JULIE 1 – 3 JULY 1987

Proceedings

of the

4th South African Computer Symposium

Holiday Inn, Pretoria

1 – 3 July 1987

edited by

Pieter Kritzinger

Computer Science Department
University of Cape Town

PREFACE

Computer science is an emerging discipline which is having difficulty in being recognised as a worthy member of the sciences. I will paraphrase John Hopcroft, co-winner of the 1986 Turing Award, when, during a recent interview, he said that the primary reason for the lack of recognition, is the age of our researchers. Probably not one of the researchers who presented their work at this symposium is older than 45. I know of no computer scientist in South Africa who is in a position where (s)he can affect funding priorities. As far as I know we have no representation on any of the committees of the Foundation for Research Development and for our Afrikaans speaking fraternity, none who is a member of the *Akademie vir Wetenskap en Kuns*. It will take time and conscious effort to establish our presence. The same is true of course for our universities. Again, with one exception, I know of no dean of a science faculty, vice-principal or principal who is a computer scientist. We consequently spend an enormous amount of time trying to explain the needs of computer science and its difficulties. I believe this symposium is a further step towards accreditation by our peers and superiors from the other sciences.

The total number of papers submitted to the Programme Committee for consideration was 34. Each paper was reviewed by three persons knowledgeable in the field it represents. Of those submitted, 23 were finally selected for inclusion in the symposium. As a result the overall quality of the papers is high and as a computer science community in Africa we can be justly proud of the final programme.

This is the fourth in the series of South African computer symposia. This year the symposium is sponsored by the Computer Society of South Africa (CSSA), the South African Institute for Computer Scientists and the local IFIP Committee. The executive director of the CSSA and his staff deserve warm thanks for handling the organisation as well as they have, while the Organising Committee provided Derrick and I with very valuable advice.

Finally I would like to express my sincere appreciation to the authors, to the members of the Programme Committee and particularly the reviewers. Without the kind cooperation of everyone, this symposium would not have taken place.

Pieter Kritzinger
July 1987.

SYMPOSIUM CHAIRMAN: PS Kritzinger, University of Cape Town.

SYMPOSIUM CO-CHAIRMAN: D Kourie, University of Pretoria.

MEMBERS OF THE PROGRAMME COMMITTEE

Judy Bishop, Witwatersrand University

Chris Bornman, UNISA

Hannes de Beer, Potchefstroom University

Gideon de Kock, Port Elizabeth University

Jaap Kies, Western Cape University

Derrick Kourie, Pretoria University

Pieter Kritzinger, Cape Town University

Tony Krzesinski, Stellenbosch University

Michael Laidlaw, Durban Westville University

Peter Lay, Cape Town University

Ken MacGregor, Cape Town University

Theo McDonald, Orange Free State University

Jan Oosthuizen, University of the North

Dennis Riordan, Rhodes University

Alan Sartori-Angus, Natal University

John Shochot, Witwatersrand University

Theuns Smith, Rand Afrikaans University

Trevor Turton, ISM (Pty) Ltd

Gerrit Wiechers, Infoplan.

LIST OF REVIEWERS

BERMAN Sonia
BISHOP Judy
BORNMAN Chris
CAREY Chris
CHERENACK Paul
DE BEER Hannes
DE VILLIERS Pieter
GORRINGE Pen
KIES Jaap
KOURIE Derrick
KRITZINGER Pieter
KRZESINSKI Tony
LAIDLAW Michael
LAY Peter
MacGREGOR Ken
MATTISON Keith
McDONALD Theo
RENNHACKKAMP Martin
RIORDAN Denis
SATORI-ANGUS Alan
SCHOCHOT John
SMITH Theuns
TURTON Trevor
VAN DEN HEEVER Roelf
VAN ROOYEN Hester
VON SOLMS Basie
VOS Koos

TABLE OF CONTENTS

Keynote Address

"An Extensible System and Programming Tool for Workstation Computers." 1
Niklaus Wirth, ETH, Zurich

Invited Lectures

"The Relationship of Natural and Artificial Intelligence." .....not included in Proceedings.
G Lasker, University of Windsor, Ontario.

"Software Engineering: What Can We Expect in the Future?" .....not included in Proceedings.
D Teichrow, University of Michigan, U.S.A.

Computer Languages I

"SPS-Algol: Semantic Constructs for a Persistent Programming Language." 13
S Berman, University of Cape Town.

"Petri Net Topologies for a Specification Language." 25
R Watson, University of the Witwatersrand.

"Towards a Programming Environment Standard in LISP." ... 45
R Mori, University of Cape Town

"ADA for Multiprocessors: Some Problems and Solutions.".. 63
J Bishop, University of the Witwatersrand.

Computer Graphics

"Polygon Shading on Vector Type Devices." 75
C F Scheepers, CSIR.

"Hidden Surface Elimination in Raster Graphics Using Visigrams." 97
P Gorringe, CSIR.

Database Systems I

"On Syntax and Semantics Related to Incomplete Information Databases." 109
M E Orlowska, UNISA.

"Modelling Distributed Database Concurrency Control Overheads." 131
M H Rennhackkamp, University of Stellenbosch.

Operating Systems

"The Development of a Fault Tolerant System for a Real-time Environment." 149
M Morris, CSIR.

"A New General-purpose Operating System." 161
B H Venter, CSIR.

Computer Languages II

"The Representation of Chemical Structures by Random Context Structure Grammars." 175
E M Ehlers and B von Solms, RAU.

"A Generalised Expression Structure." 189
W van Biljon, CSIR.

Computer Networks and Protocols I

"An Approximate Solution Method for Multiclass Queueing Networks with State Dependent Routing and Window Row Control." 203
A E Krzesinski, University of Stellenbosch.

"A Protocol Validation System." 227
J Punt, University of Cape Town.

Computer Networks and Protocols II

"Protocol Performance Using Image Protocols." 251
P S Kritzinger, University of Cape Town.

Artificial Intelligence

"A Data Structure for Exchanging Geographic Information." 267
A Cooper, CSIR.

"The Design and Use of a Prolog Trace Generator for CSP." 279
D G Kourie, University of Pretoria.

Database Systems II

"An Approach to Direct End-user Usage of Multiple Databases." 297
M J Phillips, CSIR.

"A Semantic Data Model Approach to Logical Data Independence." 329

S Berman, University of Cape Town.

Information Systems

"The ELSIM Language: an FSM-based Language for the ELSIM SEE." 343
L du Plessis and C Bornman, UNISA.

"Three Packaging Rules for Information System Design." . 363
J Mende, University of the Witwatersrand.

Computer Languages III

"Experience with a Pattern-matching Code Generator." ... 371
M A Mulders, D A Sewry and W R van Biljon, CSIR.

"Set-oriented Functional Style of Programming." 385
C Mueller, University of the Witwatersrand.

Tutorial

The use of Modula-2 in Software Engineering." 399
N Wirth, ETH, Zurich.

DAY 1

07h30	Registration and Coffee.
08h45	Welcoming address, President of the South African Institute of Computer Scientists, Dr. G. Wiechers.
09h00	Invited Lecture. Professor D. Teichrow, University of Michigan. *Software Engineering, ... What Can We Expect in the Future.*
10h00	COFFEE
	Computer Languages I. Chairman: G. Wiechers.
10h15	S. Berman, University of Cape Town. *SPS-Algol: Semantic Constructs for a Persistent Programming Language.*
10h50	A. Watson, University of the Witwatersrand. *Petri Net Topologies for a Specification Language.*
11h25	R. Mori, University of Cape Town. *Towards a Programming Environment Standard in USP.*
11h50	J. Bishop, University of the Witwatersrand. *ADA for Multiprocessors: Some Problems and Solutions.*
12h30	LUNCH

	Computer Graphics. Chairman: D. Kourie	**Operating Systems.** Chairman: K. MacGregor.
14h00	C. F. Scheepers, CSIR. *Polygon Shading on Vector Type Devices.*	M. Morris, UNISA. *The Development of a Fault Tolerant System for a Real-time Environment.*
14h35	P. Gorringe, CSIR. *Hidden Surface Elimination in Raster Graphics Using Visigrams.*	B. H. Venter, CSIR. *A New General-purpose Operating System.*
15h15	COFFEE	COFFEE
	Database Systems I. Chairman: B. von Solms.	**Computer Languages II.** Chairman: J. Bishop.
15h30	M.E. Orlowska, UNISA. *On Syntax and Semantics Related to Incomplete Information Databases.*	E.M. Ehlers and B. von Solms, Randse Afrikaanse Universiteit. *The Representation of Chemical Structures by Random Context Structure Grammars.*
16h05	M.H. Rennhackkamp, Stellenbosch University. *Modelling Distributed Database Concurrency Control Overheads*	W. van Biljon, CSIR. *A Generalised Expression Structure.*

18h00	Cocktail Party in Cullinan Room A.

DAY 2

08h30 Keynote Address by Profesor Niklaus Wirth, Swiss Federal Institute for Technology, Zurich.
An Extensible System and a Programming Tool for Workstation Computers.

Computer Networks and Protocols I. Chairman: P.S. Kritzinger.

09h30 A.E. Krzesinski, University of Stellenbosch.
An Approximate Solution Method for Multiclass Queueing Networks with State Dependent Routing and Window Flow Control.

10h05 J. Punt, University of Cape Town.
A Protocol Validation System.

10h30 COFFEE

Computer Networks and Protocols II. Chairman: R. van der Heever.

11h00 P.S. Kritzinger, University of Cape Town.
Protocol Performance using Image Protocols.

11h35 Invited Lecture by Professor G. Lasker, University of Windsor, Ontario.
The Relationship of Natural and Artificial Intelligence.

12h30 LUNCH

	Artificial Intelligence. Chairman: G. Lasker.	**Information Systems.** Chairman: D. Teichrow.
14h00	A. Cooper, CSIR *A Data Structure for Exchanging Geographic Information.*	L. du Plessis and C. Bornman, UNISA. *The ELSIM Language: an FSM-based Language for the ELSIM SEE.*
14h35	A. I. Newcombe, University of Cape Town and R. Rada, National Library of Medicine, Maryland. *Strategies for Automatic Indexing and Thesaurus Building.*	J. Mende, University of the Witwatersrand. *Three Packaging Rules for Information System Design.*
15h15	COFFEE	COFFEE
	Database Systems II. Chairman: C. Bornman.	**Computer Languages III.** Chairman: N. Wirth.
15h30	M.J. Philips, CSIR. *An Approach to Direct End-user Usage of Mutiple Databases.*	W. van Biljon, CSIR. *Experience with a Pattern-matching Code Generator.*
16h05	S. Berman, University of Cape Town. *A Semantic Data Model Approach to Logical Data Independence.*	C. Mueller, University of the Witwatersrand. *Set-oriented Functional Style of Programming.*

16h45 Open Forum with professors G. Lasker, D. Teichrow and N. Wirth.
Moderator: Dr. D. Jacobson.

19h30 Symposium Banquet in Cullinan Room.
Guest speaker, Dr. D. Jacobson, Group Executive: Technology, Allied Technologies Limited.

DAY 3

08h00 Registration (Tutorial only).

08h30 Tutorial.
The Tutorial will be given by professor Niklaus Wirth, Division of Computer Science, Swiss Federal Institute of Technology, Zurich.

The use of Modula-2 in Software Engineering.
Topics to be covererd include:

What is Software Engineering?
Data types and structures.
Modularization and information hiding.
Definition and implementation parts.
Separate compilation with type checking.
Facilities to express concurrency.
Pompous programming style.
What could be excluded?

12h15 **Close of Symposium.**

12h30 LUNCH

POLYGON SHADING ON VECTOR TYPE DEVICES

C F Scheepers

Computer Science Division
National Research Institute for Mathematical Sciences
CSIR, P O BOX 395, Pretoria, 0001, South Africa

ABSTRACT

A method is presented whereby the interior of boundary-defined regions on vector type devices may be shaded using regular line, cross-hatch and dot patterns. Different shades are realized by changing shading parameters such as line width, distance between consecutive lines or dots, and orientation of patterns.

The algorithm partitions any polygon (not necessarily convex), into mutually exclusive, pseudo-monotone polygons that can then be shaded independently using a fast procedure. The partitioning process is based on the topology of the polygon, and notions such as type-1 and type-2 critical points are introduced.

Shading is frequently used in computer graphics applications, such as cartography, engineering graphics, art, animation and hidden-line removal.

KEYWORDS AND PHRASES

Computer graphics, shading, filling, monotone polygons, cartography, map-making.

INTRODUCTION

The shading of regions with a regular pattern of lines, dots or symbols is a task common to many applications in computer graphics. Examples of applications are choropleth maps in cartography [Crom84], cross-sections in engineering drawings, scene enhancement in animation, and light intensity approximation after hidden-line removal in three-dimensional graphics.

The outer and possible inner boundaries of regions are often represented by one or more simply-connected polygons. These polygons have non-self-intersecting edges connecting a sequence of outline coordinates. The sequence of outline coordinates (or vertices) is ordered in such a way that the significant region always lies to the same side of the edges that connect consecutive vertices. Following this convention and for the sake of convenience, the vertices on the outer boundary of a region are henceforth ordered in a clockwise direction. If a region contains holes or islands, it has inner boundaries. The vertices on inner boundaries should then be ordered in a counterclockwise direction in accordance with the convention (figure 1).

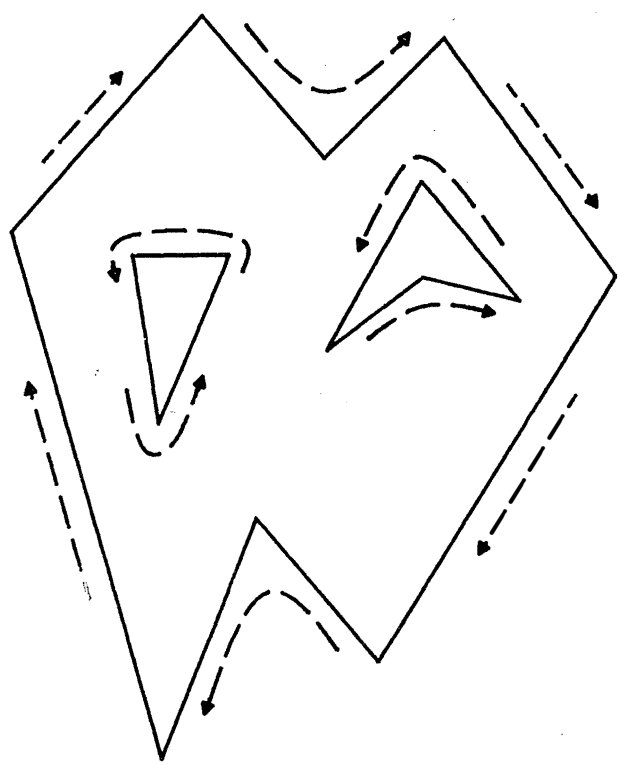


Figure 1 : A region with islands

The task at hand is to shade regions with sets of regular patterns, typically line, cross-hatch and dot patterns. The appearance of shading patterns is affected by changing input shading parameters such as line width and colour, distance between consecutive parallel lines, and angle of shading lines. Consequently, a dense pattern could be constructed that would appear darker than one with less density.

The literature on region shading (filling) in a raster type environment is extensive (see [Bras79]). The well-known polygon filling technique of scan-converting boundary-defined polygons and extentions to this technique, are discussed in [Newm79, Fole82, Gay85, Shin86]. Although adaptable to a vector type environment as in [Cart85], these techniques are often not tailored to the whims of specific requirements in vector based systems. Brassel and Fegeas [Bras79] note that cartographic applications such as map-making (typically requiring thousands of polygons with hundreds of vertices) can barely be realized successfully using this traditional approach. The reasons are twofold. Firstly, traditional algorithms are highly sensitive to the topology of regions and the density of shade lines, and secondly, these algorithms often use extensive extra storage and have a high order of complexity.

On vector type devices, a different approach to the shading problem is neccesary to enable efficient utilization of available devices. Brassel and Fegeas present a very elegant algorithm that handles shading in a way similar to the filling of an arbitrarily shaped container with water, using an influx pipe at the lowest point of the container (figure 2). Water will fill the container in the order of the labelled volumes. This analogy illustrates an approach to partition a polygon into more manageable subparts prior to shading. Lee [Lee81] employs a similar technique, except that partitioning is more efficient even though a greater number of subparts result. Cromley [Crom84] also refers to the Brassel-Fegeas algorithm and improves on the speed of the partitioning algorithm. For producing sparse shading patterns,

Cromley's technique is more efficient than the previously mentioned algorithms.

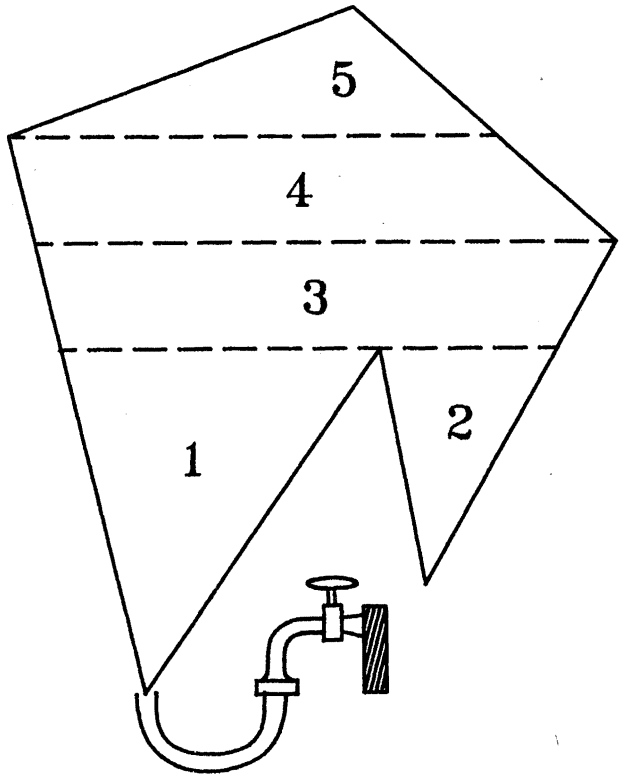


Figure 2 : Order of shading in the Brassel-Fegeas algorithm

The shading technique presented here is similar to the approach taken in these papers. A new partitioning algorithm extending on previous ideas and introducing some new concepts is presented. This algorithm yields less subparts after the partitioning process than the previous algorithms, resulting in faster shading of regions.

TERMINOLOGY

A simply-connected polygon P is represented by an ordered list of its n vertices (V_0, V_1,..., V_{n-1}) and consists of a circular sequence of edges

(V_i, V_{i+1}), for $i = 0,1,\ldots,n-1$ (index addition/ subtraction modulo n),

each edge being a line segment connecting consecutive vertices. The polygon will divide the plane into two regions, the interior and exterior regions. According to the convention presented in the introduction, the order of occurence of the vertices identifies whether the interior or exterior of the polygon forms part of the region to be shaded.

Consider a region R whose outer boundary is represented by polygon P_0. If this region contains x islands P_1, P_2,..., P_x to be excluded from shading, the vertices of island or inner boundaries are ordered in an opposite direction to the vertices of the outer boundary. Thus the region R is represented by a path-connected point set [Lee81] composed of simply-connected polygons P_0, P_1,... P_x, where P_0 is the outer boundary polygon and $P_1,\ldots,P_x$ are islands.

If a polygon P is considered topologically, three types of vertices can be identified with respect to the Y-axis. A vertex V_i of P is called a peak if both $y(V_{i-1})$ and $y(V_{i+1})$ are less than $y(V_i)$ and is called a pit if both $y(V_{i-1})$ and $y(V_{i+1})$ are greater than $y(V_i)$ [Lee81, Crom84]. These vertices are jointly called critical vertices, whereas vertices that are neither peaks nor pits are referred to as non-critical vertices (see figure 3a).

A polygon containing only two critical vertices, the ones with minimum and maximum y-coordinates, is called monotone [Lee81]. An example of a monotone polygon is illustrated in figure 3b. Note that for any closed polygon, there exist as many peaks as pits [Crom84].

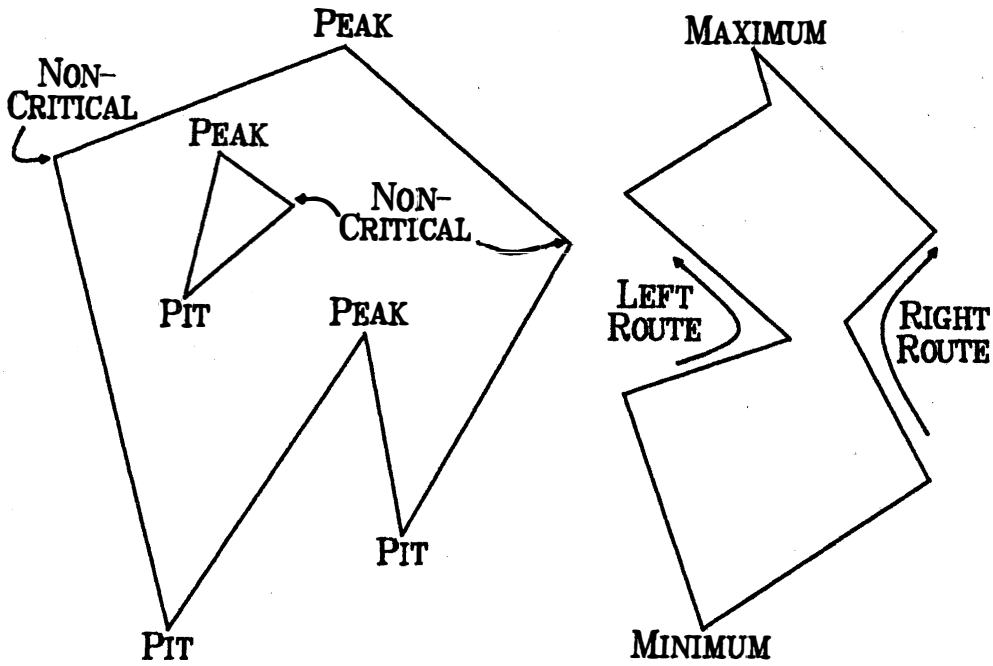


Figure 3 : (a) Peak, pit and non-critical vertices
(b) A monotone polygon

Following this terminology, a peak of type-1 is a peak of the inner region of R and a peak of type-2 is a peak of the outer region of R. Similarly, a pit of the inner region of R is called a type-1 pit and a pit of the outer region of R is a type-2 pit (figure 4).

Furthermore, a polygon is called pseudo-monotone if it contains exactly two non-crossing, non-descending routes from its minimum vertex to its maximum vertex [Sche86b]. This notion is illustrated in figure 5, where a hypothetical region has been partitioned into mutually exclusive, pseudo-monotone polygons. A pseudo-monotone polygon can be shaded in linear time.

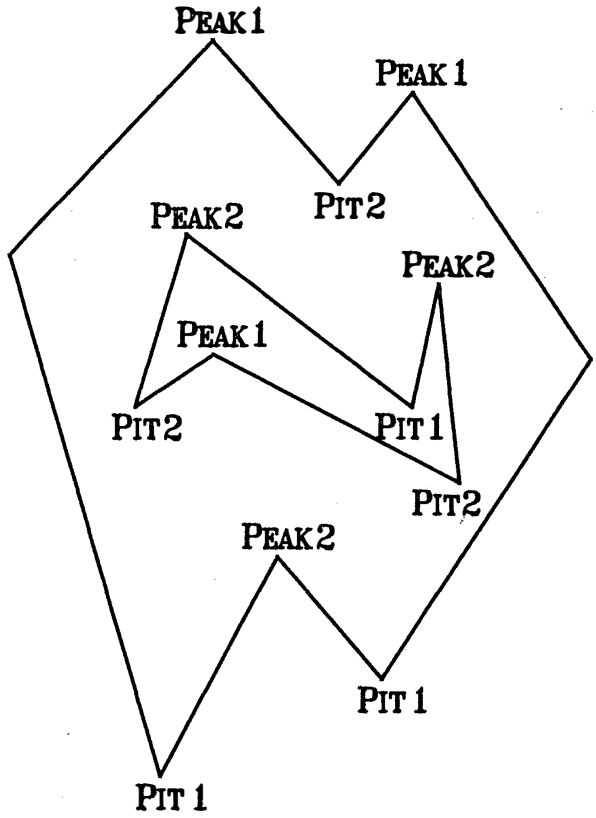


Figure 4 : Distinction between type-1 and type-2 critical vertices

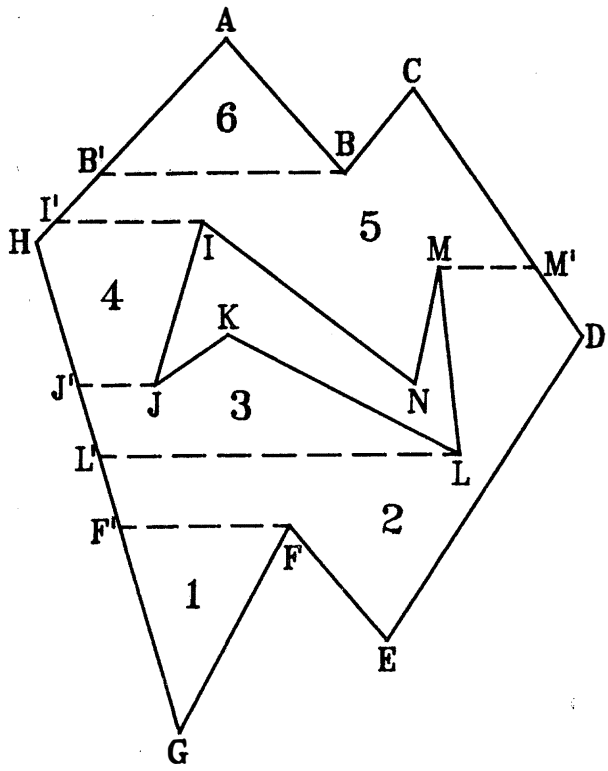


Figure 5 : Partitioning into pseudo-monotone polygons

POLYGON PARTITIONING

Assume that shading lines are parallel to the x-axis, since if they are not, the polygons $P_0,\ldots,P_x$ representing region R can always be rotated accordingly. Assume furthermore that none of the vertices have the same y-coordinates. Given these assumptions, the partitioning algorithm may now be presented.

In an initialization phase, the N vertices of all polygons $P_0\ldots P_x$ are ranked according to their y-coordinates, the vertex with minimum y-coordinate having a rank of 1. Then, all vertices of R are classified as being peaks, pits or non-critical vertices. Furthermore, during this process a distinction is made between type-1 and type-2 peaks and pits (Program A). Note that during the ranking process, vertices that have the same y-coordinates are naturally separated by assigning consecutive ranks to the vertices. Therefore, the second assumption above may be relaxed.

If region R is now considered to be a piece of paper (possibly with holes cut into it), then using a pair of scissors, it would be easy to 'cut away' pseudo-monotone polygons (pmp's) from R in the following way: Start cutting at a critical vertex of type-2 in a horizontal direction until the end of the paper in that particular direction is reached. Repeat this process for all type-2 critical vertices. The result would yield a number of oddly shaped pieces of paper that are all pmp's (Program B). Note that by using this method, the number of pmp's extracted is :

$$n(\text{pmp}) = n(\text{type-2 critical points}) + 1 - n(\text{islands})$$

As an example, consider the region illustrated in figure 5. Vertices F, L, J, M, I and B are all type-2 critical vertices. The proposed cuts are also illustrated, yielding six pmp's. The actual cutting directions chosen here is of no interest as a change in direction would still result in six pmp's.

To present the partioning process more precisely, consider the following programs:

```
PROGRAM A;
Input : A list of the N vertices of polygons P0...Px
        representing region R (vlist).
Output : Lists of ranks, classifications and types of Vi,
         rank(.), class(.) and type(.) respectively.
Functions : PRED(.) and SUCC(.) denoting the predecessor and
            successor of vertices on the boundary sequence
            respectively.
Procedures :
   - SORT(.) that sorts a list in ascending order.
   - CNODE(from, to, use, new), a procedure that creates a node
     [new] on the line through points [from] and [to], [new]
     being the horizontal cut-off point of [use] on the line.
Program block :
   BEGIN
     (* determine ranks *)
       FOR i FROM 1 TO N DO
         rank(Vi) := i;
       ENDFOR;
       SORT(rank);     (* ranks list is sorted using
                          y(Vi) as keys *)
     (* determine classifications *)
       FOR i FROM 1 TO N DO
         IF y(PRED(Vi)) < y(Vi) AND
            y(SUCC(Vi)) < y(Vi) THEN class(Vi) := peak;
         ELSIF y(PRED(Vi)) > y(Vi) AND
               y(SUCC(Vi)) > y(Vi) THEN class(Vi) := pit;
         ELSE class(Vi) := non-critical;
         ENDIF;
       ENDFOR;
```

```
(* determine types *)
  FOR i FROM 1 TO N DO
    IF class(Vi) = non-critical THEN
      type(Vi) := nil;
    ELSE
      Vi := Vi;   Succ := SUCC(Vi);
      Pred := PRED(Vi);
      CNODE(Vi, Succ, Pred, Test);     (* See figure 6 *)
      IF class(i) = peak THEN
        IF x(Test) > x(PRED(Vi)) THEN
          type(Vi) := type1;
        ELSE type(Vi) := type2;
        ENDIF;
      ELSIF class(i) = pit THEN
        IF x(Test) < x(PRED(Vi)) THEN
          type(Vi) := type1;
        ELSE type(Vi) := type2;
        ENDIF;
      ENDIF;
    ENDIF;
  ENDFOR;
ENDPROGRAM.
```

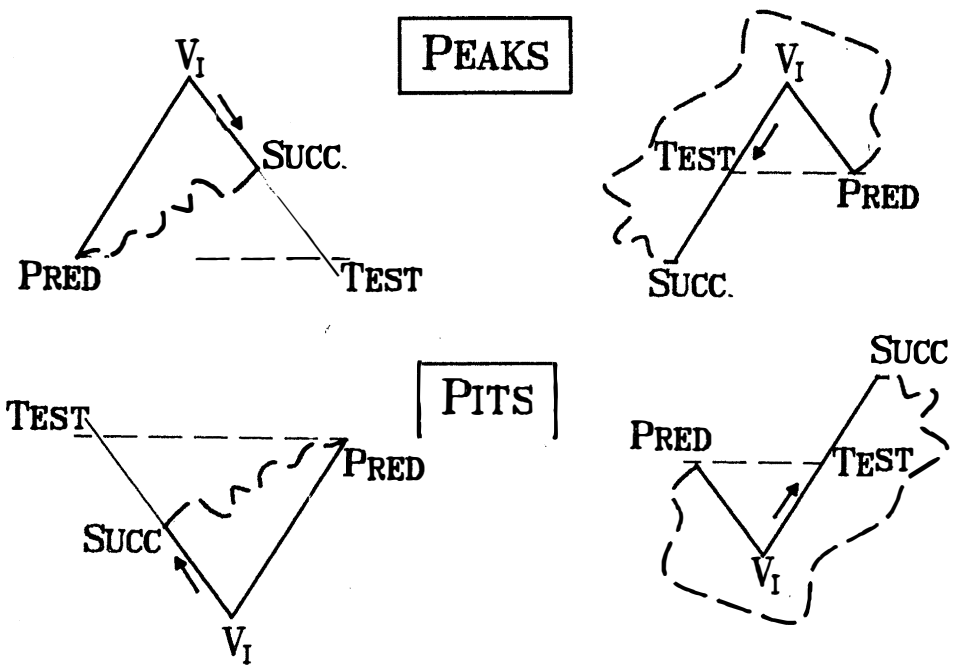


Figure 6 : Determining the types of peaks and pits

PROGRAM B;

Input :

- A list of the N vertices of polygons $P_0 \ldots P_x$ representing a region R (vlist).
- The lists of ranks, classes and types output by PROGRAM A.

Output : Two lists of vertices defining the left and right routes of a pmp (leftr and rightr respectively).

Storage :

- Two lists, one containing all type-1 pits in ascending order of ranks (plist) and the other containing all type-2 pits in ascending order of ranks (qlist).
- A list of all type-2 peaks (rlist), as well as a list of markers associated with every type-2 peak (mark(.)).
- A stack with potential maximum size s equal to the total number of type-2 pits in R, where the stack is empty initially (vstack).
- Two temporary variables (Curl and Curr) representing vertices currently under consideration.

Functions :

- MINI(a,b), returns the lower of two vertices [a] and [b].
- PRED(.), SUCC(.) and CNODE(...) as in PROGRAM A.
- TYPE2PEAK(x), true if vertex [x] is a type-2 peak.
- INLIST(list,point), true if vertex [point] is in [list].

Procedures :

- PROCEDURE ADDLIST(list, point);

```
BEGIN
  \ code to add [point] at the end of [list] \
ENDPROCEDURE;
```

- PROCEDURE CUTTOLEFT;

```
BEGIN
  (* Cut to the left edge to extract a pmp *)
  CNODE(PRED(Curl), Curl, Min, New);
  ADDLIST(leftr, New);
  mark(Min) := New;        (* Associate Min with New *)
ENDPROCEDURE;
```

```
- PROCEDURE CUTTORIGHT; (* Cut to the right edge *)
  BEGIN
    CNODE(SUCC(Curr), Curr, Min, New);
    ADDLIST(rightr, New);   mark(Min) := New;
  ENDPROCEDURE;
- PROCEDURE INSINUATE (x);
  BEGIN
    \ code to check if insinuation occurs (see figure 7)
      All elements in qlist with ranks smaller than x is
      possible insinuation points \
    IF \ insinuation occurs \ THEN
      \ code to remove the insinuation point (Inpnt) from
        qlist as a type-2 pit can 'insinuate' only once \
      CNODE(PRED(Curl), Curl, Inpnt, New);
      ADDLIST(leftr, New);    ADDLIST(leftr, Inpnt);
      \ code to push New and Inpnt onto vstack \
      Curl := SUCC(Inpnt);
    ELSE
      IF flag THEN Curl := SUCC(Curl)
      ELSE Curr := PRED(Curr);
      ENDIF;
    ENDIF;
  ENDPROCEDURE;
```

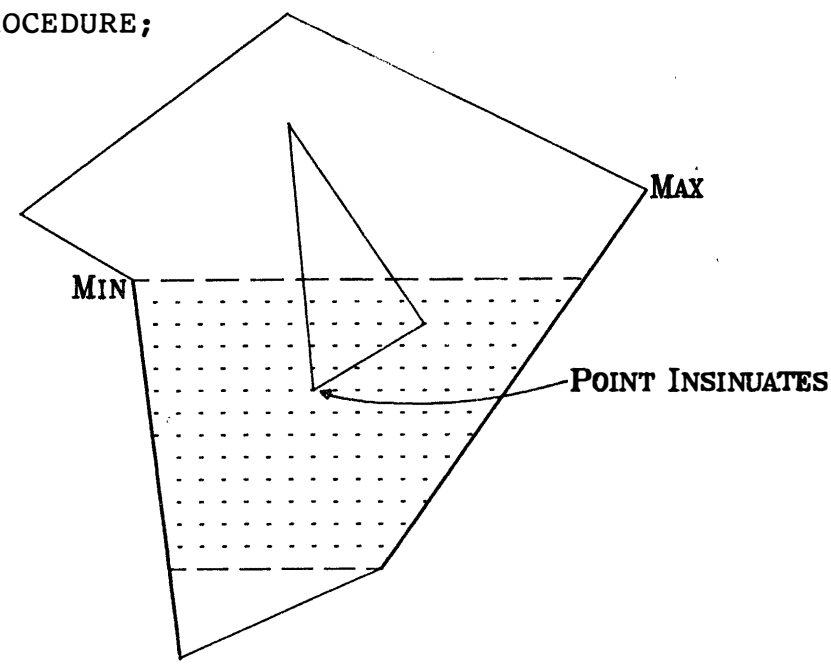


Figure 7 : An insinuation of a type-2 pit occurs if the vertex lies within the coloured area

```
- PROCEDURE XTRACT;
  BEGIN
    LOOP
      Min := MINI(Curl, Curr);
      IF Curl = Min THEN
        flag := true;    ADDLIST(leftr, Min);
      ELSE
        flag := false;   ADDLIST(rightr, Min);
      ENDIF;
      IF TYPE2PEAK(Min) THEN
        IF INLIST(rlist, Min) THEN
          \ code to remove Min from rlist \
          IF flag THEN CUTTORIGHT;
          ELSE CUTTOLEFT ENDIF;
          EXITLOOP;
        ELSE (* Peak has already been used to cut from *)
          New := mark(Min);
          IF flag THEN
            ADDLIST(leftr, New);    Curl := SUCC(New);
          ELSE
            ADDLIST(rightr, New);   Curr := PRED(New);
          ENDIF;
        ENDIF;
      ELSE (* Min is not a type-2 peak *)
        IF Curl = Curr THEN EXITLOOP
        ELSE INSINUATE(rank(Min)) ENDIF;
      ENDIF;
    ENDLOOP;
  ENDPROCEDURE;
Program block (Execute for every pmp to be extracted) :
  BEGIN
    IF NOT empty (plist) THEN
      leftr := rightr := empty;
      Curl := SUCC(plist(1));     Curr := PRED(PLIST(1));
      ADDLIST(leftr, plist(1));   ADDLIST(rightr, plist(1));
      \ code to remove the first element from plist \
      XTRACT;
    ELSIF NOT empty (vstack) THEN
      \ code to pop insinuation vertex Inpnt and point New \
      Curl := SUCC(New);  Curr := PRED(Inpnt);  XTRACT;
    ENDIF;
ENDPROGRAM;
```

POLYGON SHADING

The pseudo-monotone polygons are shaded in way very similar to the technique used in [Bras79, Crom84] for shading trapezoids or triangles. The left and right routes that have been introduced in the discussion above are used as left and right shading limits. Instead of intersecting each shading line with these limits to determine the shading line end points, an incremental displacement along each shading edge is calculated.

Assume that apart from being parallel to the x-axis, all shading lines are also placed on integer y-axis values. (If this is not true, all vertices of polygons $P_0 \ldots P_x$ can always be scaled accordingly). Consider the left route of a hypothetical pmp as illustrated in figure 8a. Let L_i denote left end points of shading lines. Then by calculating $x(L_1)$ and $y(L_1)$ once for each shading limit, with repeated additions of d_x and d_y,

$$d_x = x(L_2) - x(L_1)$$
$$d_y = y(L_2) - y(L_1) = 1 \qquad \text{(from the assumption)},$$

it is simple and fast to determine the other values of L_i : Let DX=x(B)-x(A) and DY=y(B)-y(A), then

$$y(L_1) = \text{CEILING}(y(A))$$
$$x(L_1) = x(A) + (y(L_1) - y(A))*DX/DY$$

and

$$d_x = DX/DY$$
$$d_y = 1.$$

Hence, the shading line end points are :

$$x(L_i) = x(L_{i-1}) + d_x$$
$$y(L_i) = y(L_{i-1}) + 1,$$

for i = 2...COUNT, where

COUNT = FLOOR(y(B)) - CEILING(y(A)) + 1

A similar approach is taken to determine positions on a conceptual shading line for the placement of dots or symbols [Sche86a]. Using the parameters illustrated in figure 8b (density, indentation and distance between consecutive shading lines), it is possible to construct many different dot and symbol patterns (see figures 9 and 10).

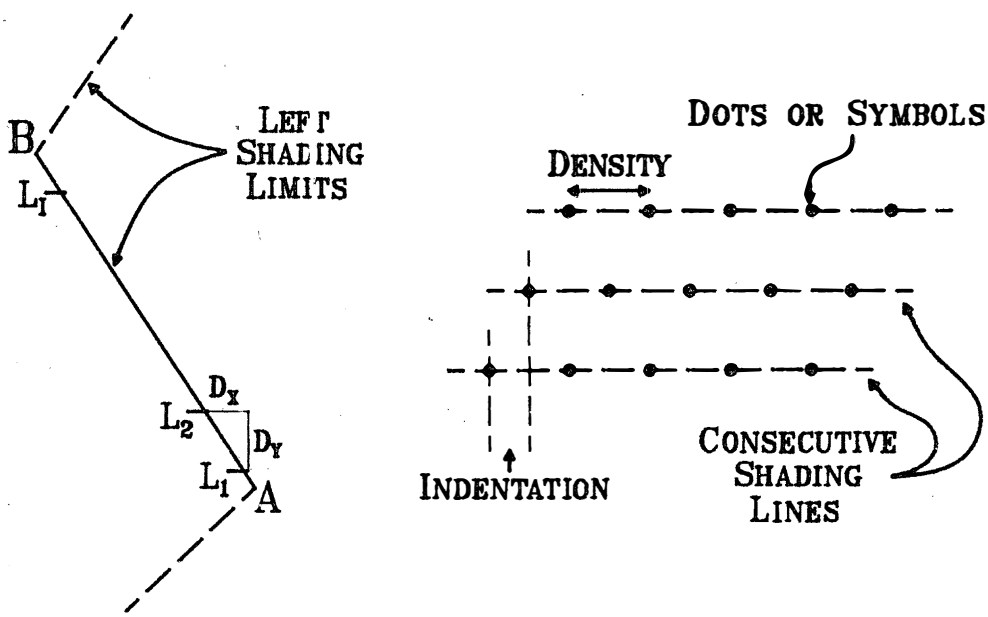


Figure 8 : (a) Determining shading line end points
(b) Parameters to control the shading using dots or symbols

IMPLEMENTATION

The shading technique described above has been implemented on a PERKIN-ELMER (3200-series) mini-computer in a locally developed procedural language called SCRAP. Using storage management facilities, it was possible to provide for regions with any number of islands and any number of vertices (only restricted by the storage available). This facility is required in a cartographic system currently being developed where geographical regions often contain a large number of islands, and the total number of vertices describing a region can be in the order of thousands.

The procedure first eliminates duplicate vertices on the boundaries of polygons representing the region, and then processes the island boundaries by reordering them in an opposite direction to the order of the outline boundary. Depending on the required parameters for shading, whether for shading lines or dot patterns, the boundary vertices are scaled and rotated so that conceptual shading line segments are always horizontal and fall on integer y-coordinates. Following this, the initialization phase requires a single O(n LOG n) sort on y-coordinates. These sorted coordinates are used to classify peak, pit and non-critical vertices of all polygons in an order O(n) process. The region to be shaded is then partitioned into pseudo-monotone polygons that can be shaded in linear time. Note that in the above discussion, no reference has been made to the fact that shading line segments should be rotated and scaled back to the original coordinate system. This is not necessary as the shading line end points can be calculated directly in the original coordinate system (see [Sche86a]).

Comparing this technique with the Brassel-Fegeas algorithm, it is obvious that the pmp partitioning algorithm is more effective. Apart from only requiring a single sort (as apposed to two sorts in the Brassel-Fegeas algorithm), this algorithm partitions a region into

N(pmp's) = N(type-2 pits and peaks) + 1 - N(islands).

The Brassel-Fegeas algorithm on the other hand, partitions the same region into N trapezoids, where

$$N(\text{trapezoids}) = N(\text{vertices}) - 1 + N(\text{islands}),$$

The reduction in the number of subparts that a partitioning process requires forms the strenght of the new algorithm as the number of subparts may be considered as an overhead of the algorithm (see [Lee81]).

Furthermore, the Brassel-Fegeas algorithm uses all outline vertices to test for insinuation, making it a very time intensive process. In the present algorithm, which uses the classifications of vertices, the test for insinuation is highly optimized as the vertices to be considered each time are limited.

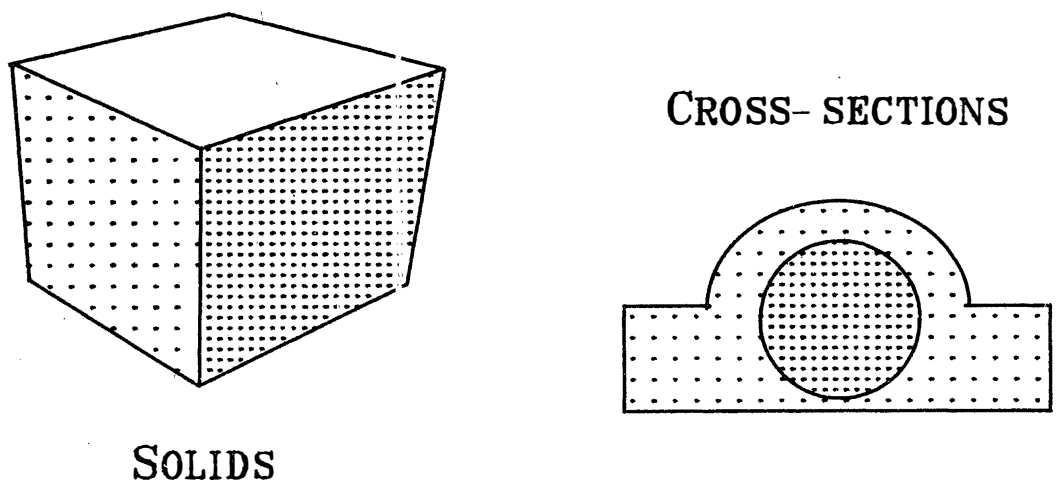
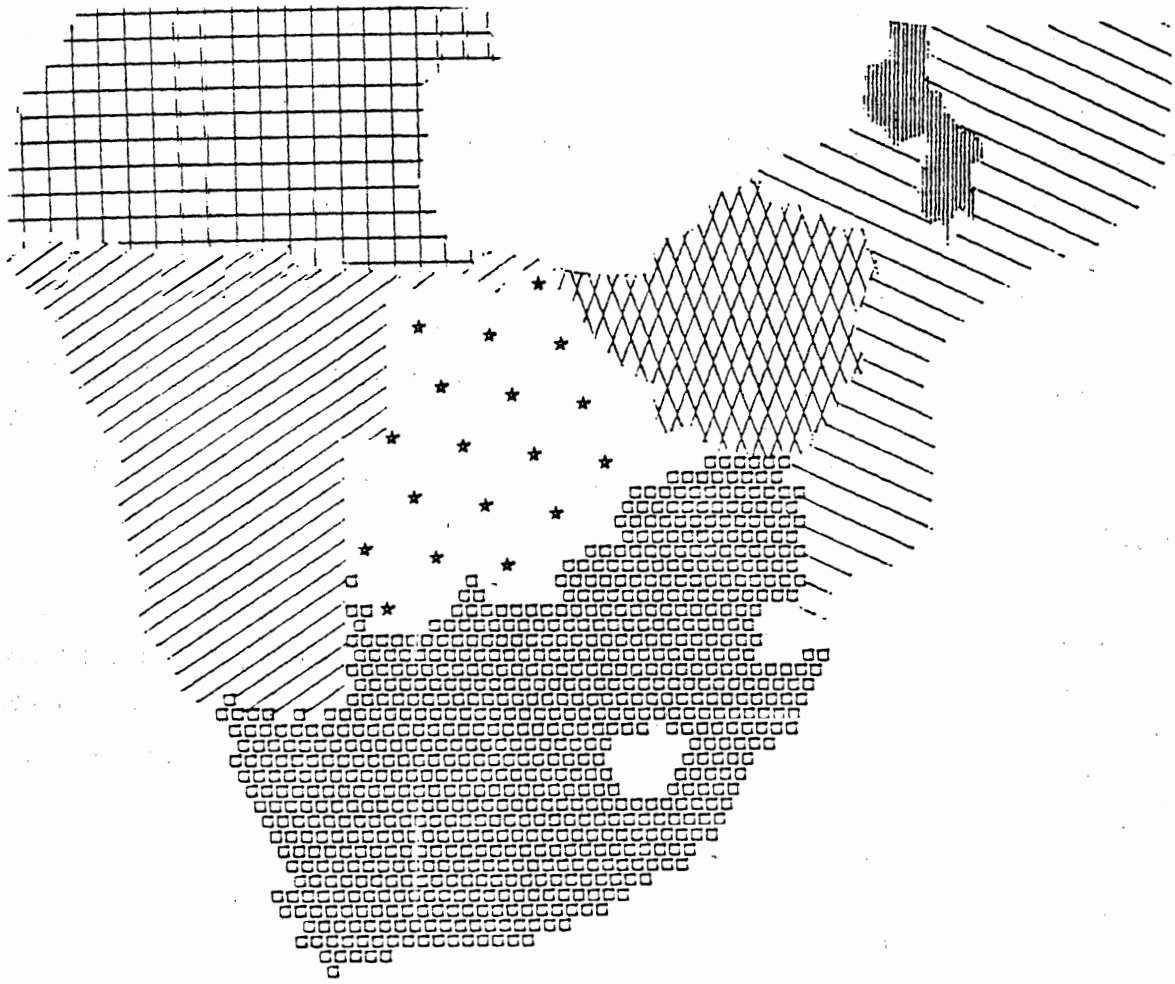


Figure 9 : Examples of different dot pattern shading



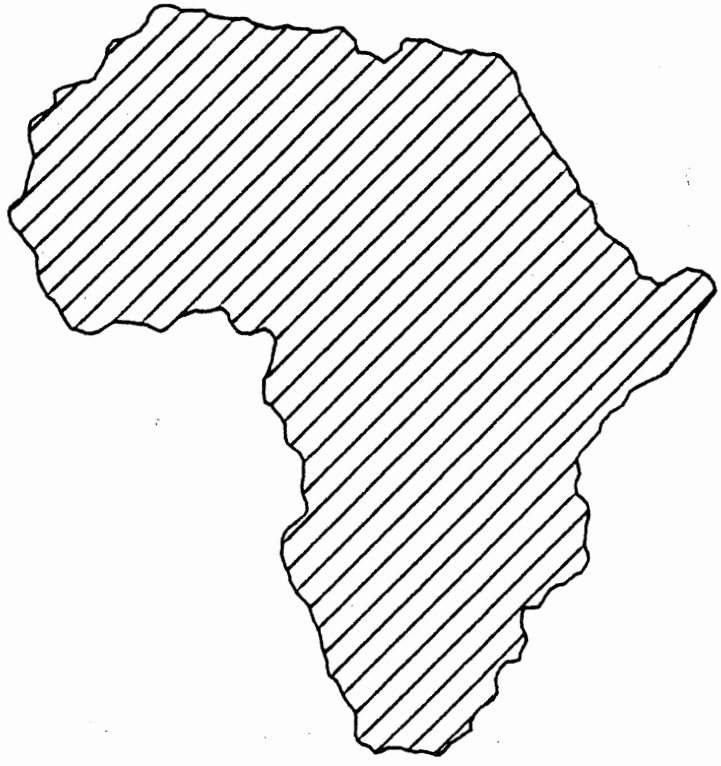


Figure 10a : Examples of results obtained with the pmp partitioning algorithm

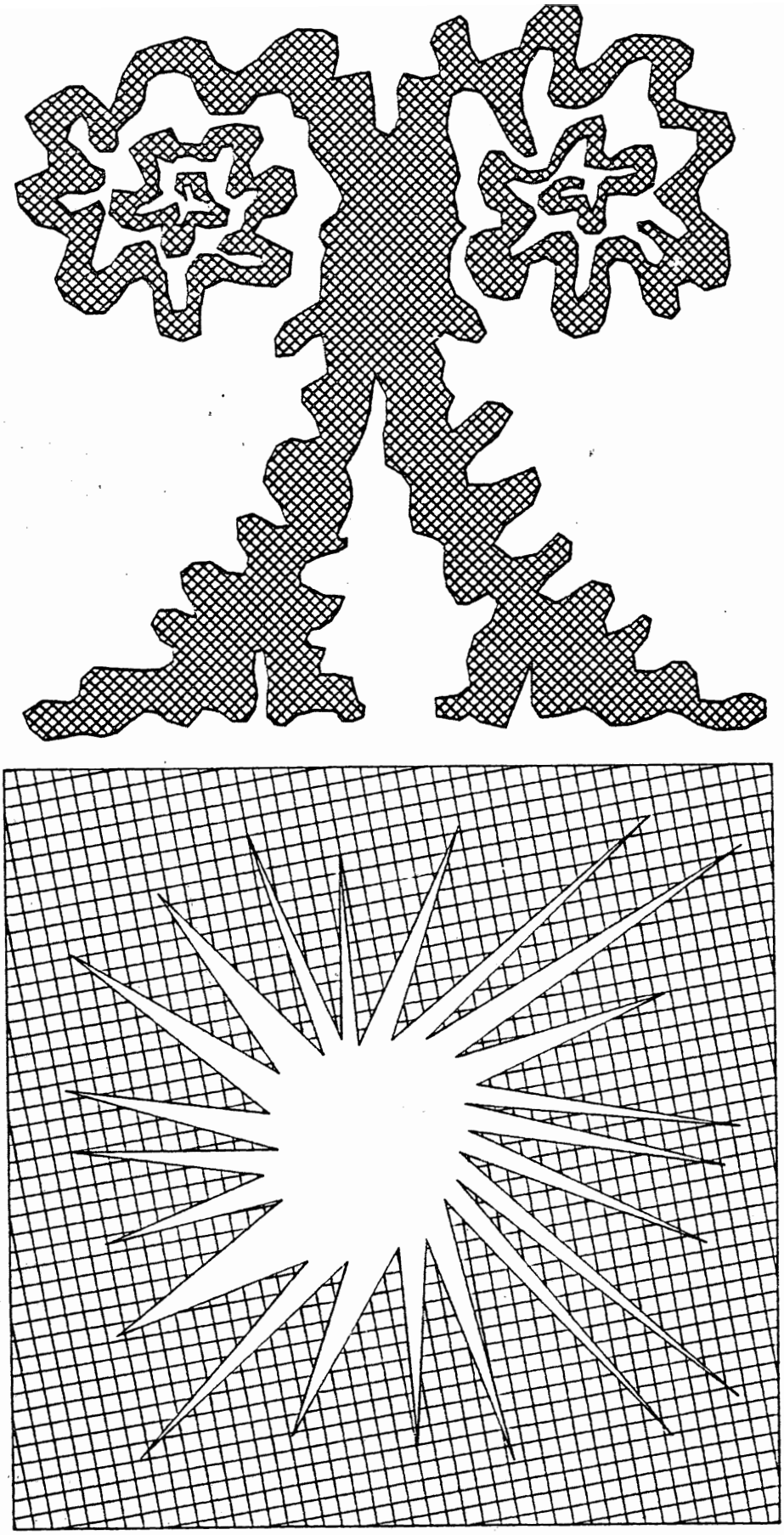


Figure 10b : Examples of results obtained with the pmp partitioning algorithm

CONCLUDING REMARKS

An efficient technique for shading regions on vector type devices has been presented. This algorithm uses the topology of the region to partition it into less complex subparts that are then shaded by means of a fast procedure.

Using shading parameters, many different regular shading patterns may be constructed, making this technique very useful in a number of different computer graphics applications. Although the algorithm has been designed specifically for vector type devices, it can also be used for shading on raster type devices, where shading lines may be viewed as scan lines.

REFERENCES

[Bras79] BRASSEL,K.E. FEGEAS,R. "An algorithm for shading of regions on vector display devices" (Computer Graphics, Vol.13, No.2, August 1979, 126-133).

[Cart85] CARTLEDGE,C.J. WEEKS,G.H. "Implementation of area fill for GKS" (In : EARNSHAW,R.A. (ed.) Fundamental algorithms for computer graphics, NATO ASI Series F: Computer and Systems Sciences Vol.17, Springer-Verlag, Berlin, 1985, 162-185).

[Crom84] CROMLEY,R.G. "The peak-pit-pass polygon line-shading procedure" (The American Cartographer, Vol.11,No.1, 1984, 70-79).

[Fole82] FOLEY,J.D. VAN DAM,A. Fundamentals of interactive computer graphics (Addison-Wesley Publishing Company, Inc., 1982, 664pp).

[Gay85] GAY.A.C. "Experience in practical implementation of boundary-defined area fill" (In : EARNSHAW,R.A. (ed.) Fundamental algorithms for computer graphics, NATO ASI algorithms for computer graphics, NATO ASI Series F: Series F: Computer and Systems Sciences Vol.17, Springer-Verlag, Berlin, 1985, 153-160).

[Lee81] LEE,D.T. "Shading of regions on vector display devices" (Computer Graphics, Vol.15, No.3, August 81, 33-44).

[Newm79] NEWMAN,W.M. SPROULL,R.F. Principles of interactive computer graphics (McGraw-Hill, Inc., 1979, 541pp).

[Sche86a] SCHEEPERS,C.F. "Veelhoekskakering in vektorgrafika" (NNWW Interne verslag, I679, Pretoria, April 1986, 37pp).

[Sche86b] SCHEEPERS,C.F. "Graphical communication and symbolism in a computerized cartography and map compilation system" (In : BRADSHAW,J. WRENCH,K. (eds.) First Computer Science research students conference, Stutterheim, June 1986, (proceedings), Dept. of Computer Science, Rhodes University, August 1986, 133-154).

[Shin86] SHINDE,Y.N. MUDUR,S.P. "Algorithms for handling the fill area primitive of GKS" (Computer Graphics Forum, Vol.5, 1986, 105-117).

[Ying85] YING,D. FENG,X. "Arbitrary area filling in a fast procedure" (Computer Graphics Forum, Vol.4, 1985, 363-370).