Comput. & Graphics Vol. 13, No. 2, pp. 193–206, 1989
Printed in Great Britain.

0097-8493/89 $3.00 + .00

Technical Section

ON THE TWO ARRAY MASK HIDDEN-LINE ALGORITHM

L. ALVISI and G. CASCIOLA
Department of Mathematics, University of Bologna, Italy

Abstract—This paper considers the Two Array Mask algorithm which enables rapid plotting of a continuous function $F(x, y)$. After reexamining this method as a hidden-surface algorithm, its correctness and precision were analyzed for parallel and perspective projections; given that the method with perspective projections is, generally speaking, not exact, alternative solutions are proposed for this case.

1. INTRODUCTION

The solution to many problems lies in the functions of two variables, the functions being given either by an explicit description or a grid of values.

In such cases the function has to be given a realistic visual representation (i.e., with the hidden lines or surfaces eliminated). The algorithms mentioned in the literature for plotting the functions of two variables are of the hidden-line type operating in image space. A surface can either be represented using perspective or parallel projections, the former creating a visual effect similar to that of the human eye or of a photographic system.

The most widely used algorithm in the literature for solving such problems is the one called the "Floating Horizon" or "Two Array Mask" (TAM), which was used by some authors for parallel representation of a function [2, 4, 5, 7, 8] while it was used by others for perspective representation [3, 6]. One of the characteristics of this method is its rapidity, as the hypotheses of a single-valued continuous function are fully exploited. With this method the surface is represented by drawing one or two families of curves, according to a strict order.

Williamson [2] solved the problem in parallel projections, by using a set direction of projection; Watkins [4] used sights taken from any direction whatsoever, yet noted that some of them led to lines being erroneously eliminated. Butland [5] optimized the method, but only by including viewing from surface domain corners, and Sowerbutts [7], expanded upon Butland's ideas by futher extending viewing to considering the front and sides. Skala [8] found evidence of a drawing order for curves which was "correct" for every direction of projection, his main contribution being a particularly interesting and efficient modification that he made to Bresenham's Drawing Line Algorithm. Wright [3] suggested further expansion upon Williamson's method to include perspective projections, but did not clearly describe the procedure for each center of projection in the drawing order of the curves.

In this paper we propose to reexamine the TAM method for hidden-line elimination as an algorithm for the elimination of hidden surfaces. It will in this sense be necessary to establish the "correct" surface-drawing order and an "exact" plotting method.

The feasibility of the TAM method for both parallel and perspective projections, for each direction and center of projection respectively, was analyzed in order to display a cross-hatched surface.

On the basis of this analysis, in the case of perspective projections, a variation on the TAM method is suggested that enables greater drawing accuracy as well as further alternatives for accurate plotting.

2. PROBLEM

Given the continuous function

$$z = F(x, y) \text{ with } (x, y) \in [a, b]x[c, d], \quad (1)$$

it has to be graphically represented with the use of a display raster. Selection has to be made from the infinite points of the function to identify a subset that will conveniently represent it. The most instinctive and efficient way of achieving this end is by considering the following rectangular grid on the domain:

$$(x_i, y_j) \quad i = 1, \ldots, NX \quad j = 1, \ldots, NY$$

with

$$x_i = a + (i - 1)hx$$

$$y_j = c + (j - 1)hy$$

where

$$hx = (b - a)/(NX - 1) \quad \text{and}$$

$$hy = (d - c)/(NY - 1)$$

thereby considering the triples $(x_i, y_j, F(x_i, y_j))$. From this finite number of points it is necessary to construct a function, which we will call $F^*(x, y)$, definite and continuous on the domain of (1), that approximates $F(x, y)$.

$F^*(x, y)$ can be obtained rapidly at an arbitrary point by selecting a linear surface patch approximation.

Let each pair of subscripts (i, j), with $1 \le i \le NX - 1$ and $1 \le j \le NY - 1$, be associated with the natural number, k, obtained from:

$$k = (j - 1)(NX - 1) + i.$$

This research has been supported by the Consiglio Nazionale delle Ricerche—Scienze Matematiche (C.N.R., Italy) under Contract no. 87.00997.01.

The rectangular grid unit with vertices (x_i, y_j), (x_i, y_{j+1}), (x_{i+1}, y_{j+1}), (x_{i+1}, y_j) may, therefore, be defined as E_k (Fig. 1).

As it is known that three non-aligned points will determine a plane in space, it generally follows that a linear approximation that will satisfy (1) in the four vertices of E_k does not exist. However, by selecting three of the four points, linear approximation can be obtained that is feasible for half of E_k. The remaining half can be approximated in like fashion.

This can, of course, be done in two different ways, each of them resulting in slightly different approximations of (1). In order that this should have no effect upon final representation, (1) has to be sufficiently regular.

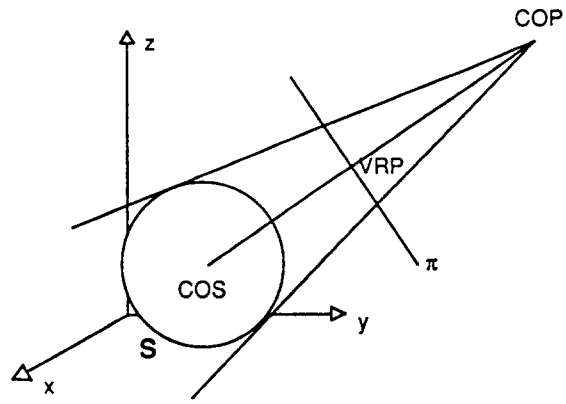

Fig. 2.

3. PROJECTIONS

There are two principal classifications of geometric plane projections: parallel and perspective ones. The distinction between the two is to be found in the relation between projection center and projection plane. If the distance between the two is finite, projection will be perspective, whereas if the distance is infinite, projection will be parallel. Perspective projections are categorized by the number of principal vanishing points they have and therefore by the number of axes the projection plane cuts.

The following general scheme has been adopted in this paper: Let COS be the center of the smallest sphere, S, containing the triples $(x_i, y_j, F(x_i, y_j))$ for $i = 1, \ldots, NX, j = 1, \ldots, NY$.

For *perspective projections* a center of projection (COP) is selected outside the sphere S with the aim of seeing the surface in its entirety. As projection plane, we define π to be orthogonal to the COP and COS endpoint segment. In particular, we set π to cut the segment in its mean point; we call VRP this point of intersection (Fig. 2).

An X-Y orthogonal cartesian reference system is defined upon π, with origin at VRP, and Y-axis consisting of a VUP vector projection. Here VUP is parallel to the z-axis (if the COP is selected so that COP–COS is parallel to the z-axis, the VUP will be selected parallel to the y-axis). Finally, the X-axis is selected so that a left-hand system is formed by the X and Y axes and the COP–COS vector.

A squared window containing the projection of sphere S, is then defined upon π (Fig. 3).

For *parallel projections*, we have to select VRP. The direction of projection will be the one set by the vector VRP–COS, where, without changing general cases, VRP is selected outside the sphere S (Fig. 4).

As we did with perspective projections, we define the projection plane π to be orthogonal to VRP–COS and passing through VRP; we also define, an X-Y reference system upon it and a squared window.

(X, Y) will subsequently be used to indicate the coordinates of a point upon π and (XS, YS) to indicate its device coordinates.

4. GRID PATH

Let us say that $F^*(E_k)$ is the approximated F^* function restricted to the E_k grid unit; therefore F^* may be considered on the given domain as the union of the $F^*(E_k)$.

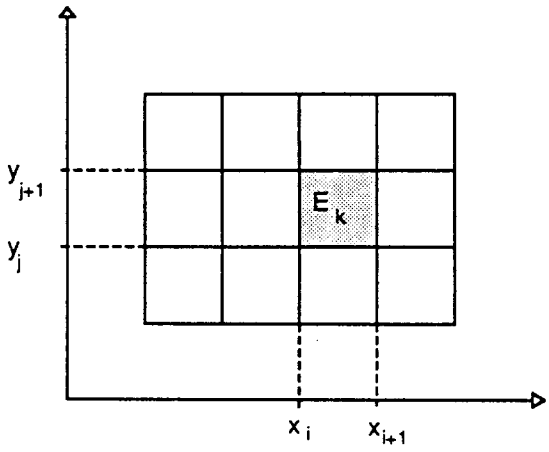

Fig. 1.

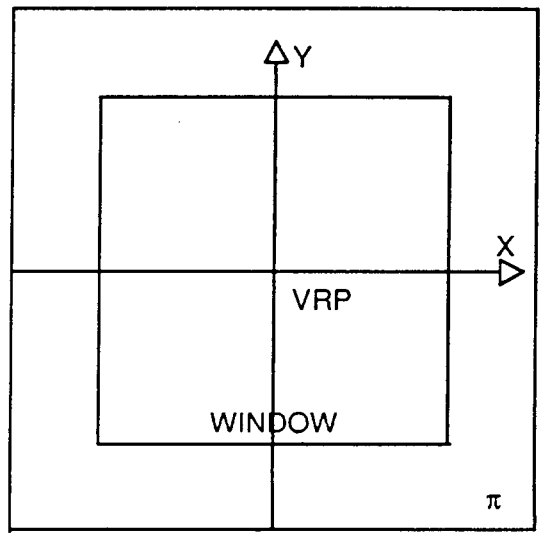

Fig. 3.

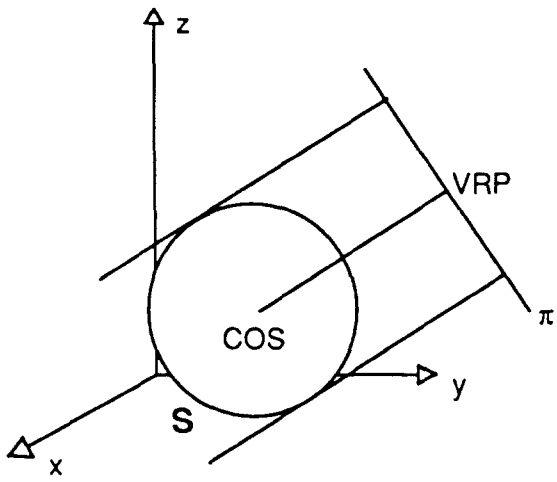

Fig. 4.

Sutherland, Sproull and Shumaker[9] had found evidence that the elimination of hidden surfaces could be considered as an enormous sorting process. It may be observed that because F, and consequently F^*, are single-valued continuous functions, the $F^*(E_k)$ sorting with respect to a center or direction of projection can be obtained with low computational complexity.

We need some kind of $F^*(E_k)$ sorting so that:

> if $F^*(E_k)$ occupies position i in the sorting, then it is either visible or (partially or totally) hidden by $F^*(E_k)$ surface units occupying previous positions to the i-th (in order ranging from the nearest to the furthest). (2)

Definition. A grid path is said to be any kind of ordered succession of all E_k units.

Definition. A grid path is said to be correct if condition (2) is satisfied by the succession of E_k units.

Definition. Two $F^*(E_k)$ and $F^*(E_h)$ surface units are said to be p-independent if there is no straight projection ray (projector) p intersecting either of them. Whereas they are said to be p-dependent if such a projector does, in fact, exist.

Definition. Two $F^*(E_k)$ and $F^*(E_h)$ surface units are said to be α-independent, if no plane α, containing a projector p and orthogonal to x-y plane, intersects both of them. Whereas they are said to be α-dependent if such a plane does, in fact, exist.

The edges of $F^*(E_k)$ do belong to planes which are perpendicular to the x-y plane and orthogonal to each other. The order in which a plane α that is perpendicular to the x-y plane cuts these orthogonal planes, can be estimated to be the same as the order in which line r, intersection of α and x-y, cuts the definite grid on the domain.

On account of the existing bijection between units $F^*(E_k)$ and E_k, it follows that line r's order of intersection with the E_k grid units can be considered to be the same as the plane α's order of intersection with the $F^*(E_k)$ surface units.

This means putting the α-dependent $F^*(E_k)$ units in an ordering. Note that two p-independent surface units are also α-independent, whereas it does not necessarily follow that two α-dependent units are also p-dependent.

It can therefore be deduced that considering an order among α-dependent $F^*(E_k)$ units is more restrictive than it is among p-dependent units. Yet this enables an order to be established merely upon the basis of the grid units, and independently of the z-coordinates of the surface units, i.e., of F's z-coordinates at the grid points.

Problem

Let us say that E_k grid units, $k = 1, \ldots, (NX - 1)(NY - 1)$ are assigned upon the $[a, b] \times [c, d]$ rectangular domain.

Given a COP for perspective projections or a VRP for parallel projections, the correct grid path can be determined.

It can above all be observed that the COS projection onto x-y plane will fall at the center of the domain, in other words $\mathrm{COS}x \equiv (b - a)/2$ and $\mathrm{COS}y \equiv (d - c)/2$. Therefore: for *parallel projections* a line s can be indicated, which does not interesect the domain, and is orthogonal to the ray with origin at $(\mathrm{COS}x, \mathrm{COS}y)$ and passing through $(\mathrm{VRP}x, \mathrm{VRP}y)$. r_l rays parallel to $(\mathrm{COS}x, \mathrm{COS}y) - (\mathrm{VRP}x, \mathrm{VRP}y)$ may, therefore, be considered to have origins on line s and to intersect the domain (Fig. 5).

For *perspective projections* r_l rays may be considered to have origins on $(\mathrm{COP}x, \mathrm{COP}y)$ and to intersect the domain (Fig. 6).

Let us say that each r_l ray induces a local order of the E_k grid units it intersects, given by the very intersection order of these units themselves.

The local ordering induced by an r_l ray can be indicated by listing the n_l subscripts of the grid units intersected

$$(k_1, k_2, \ldots, k_{n_l}).$$

Let P be the set of all the permutations of the first

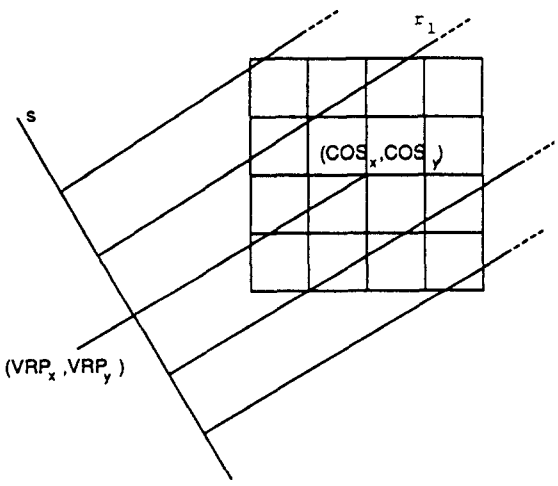

Fig. 5.

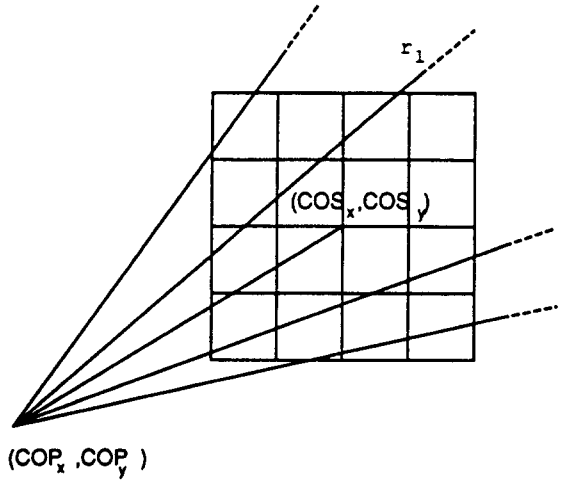

Fig. 6.

natural numbers $(NX - 1)(NY - 1)$, and therefore all possible grid paths:

$$P = \{p_1, p_2, \ldots, p_{(NX-1)(NY-1)}\}$$

where

$$p_h \equiv (k_{h,1}, k_{h,2}, \ldots, k_{h,(NX-1)(NY-1)})$$
$$1 \leq k_{h,i} \leq (NX - 1)(NY - 1)$$

An r_l ray will reduce all the possible P paths to only those of:

$$P_{r_l} = \{p_h; \text{ if } u < v \text{ and } k_u = k_{h,i} \text{ and } k_v = k_{h,j}$$
$$\text{then } i < j \text{ for each } u, v = 1, \ldots, n_l\}.$$

In other words P_{r_l} is the set of the p_h containing units $k_1, k_2, \ldots, k_{n_l}$ in the order induced by r_l.

Let all and only the rays r_l which intersect the domain and induce different local sortings among E_k grid units be m; it follows therefore that all the correct paths possible can be derived from

$$\bigcap_{l=1,\ldots,m} P_{r_l}.$$

Example

Let P be the set of all the permutations of numbers 1, 2, 3 and 4, i.e., $P = \{(1, 2, 3, 4), (1, 2, 4, 3), \cdots\}$ (cfr. Fig. 7).

r_1 cuts the grid units in the following order:

$$E_1, E_2, E_4$$

therefore

$$P_{r_1} = \{(1, 2, 3, 4), (1, 3, 2, 4),$$
$$(1, 2, 4, 3), (3, 1, 2, 4)\}.$$

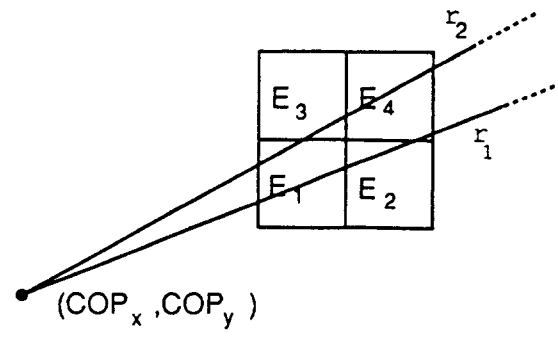

Fig. 7.

r_2 cuts the grid units in the following order:

$$E_1, E_3, E_4$$

therefore

$$P_{r_2} = \{(1, 2, 3, 4), (2, 1, 3, 4),$$
$$(1, 3, 2, 4), (1, 3, 4, 2)\}.$$

and therefore the correct paths are:

$$\bigcap_{l=1,2} P_{r_l} = \{(1, 2, 3, 4), (1, 3, 2, 4)\}.$$

For parallel projections it can be observed that working out the (x_i, y_j) grid point nearest to the line s, is the same as determining the minimum of a linear function constrained by a rectangular domain. According to the theory of Linear Programming, the minimum is definitely a vertex of the domain. From this it can be deduced that among the correct paths there will be at least one (usually all of them) which will have as first element a corner grid unit (Fig. 8).

Whereas in the case of perspective projections, the first grid unit in each correct path will not usually be a corner unit; in fact the problem is the same as identifying the minimum of a convex function that is constrained to a rectangular domain.

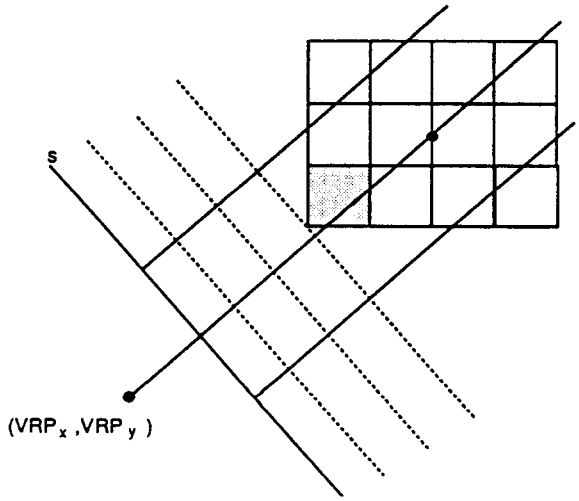

Fig. 8.

All the possible cases are shown in Figs. 9a–c.

According to the theory of Convex Programming there is only one global minimum, but it is not usually on one of the vertices of the domain, which, on the other hand, is the case if the maximum is being sought (in which case there would be some local maxima, but these would, in fact, be on the vertices of the domain).

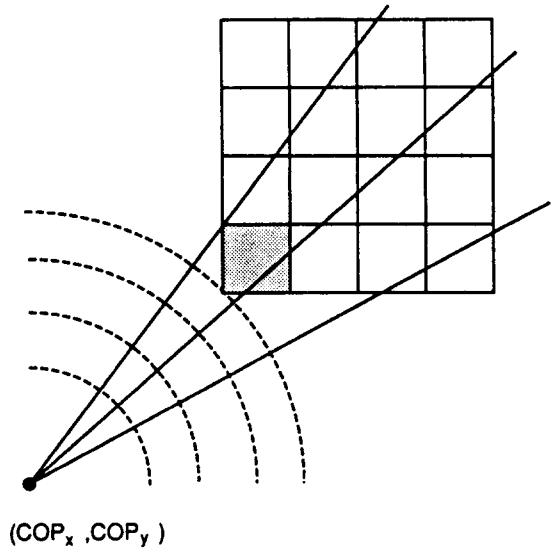

Fig. 9a.

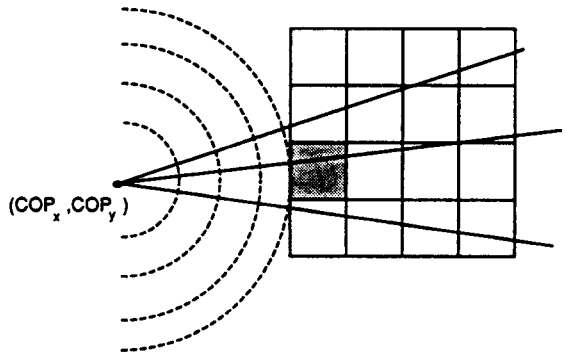

Fig. 9b.

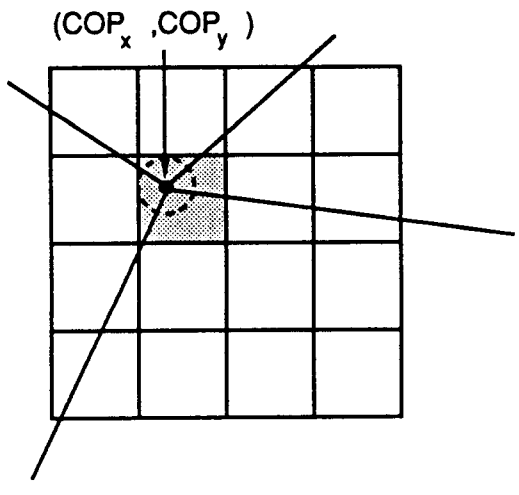

Fig. 9c.

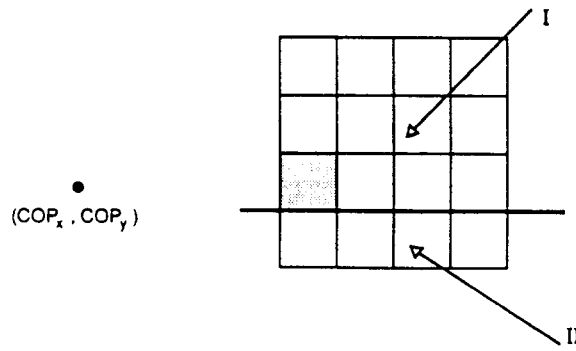

Fig. 10a.

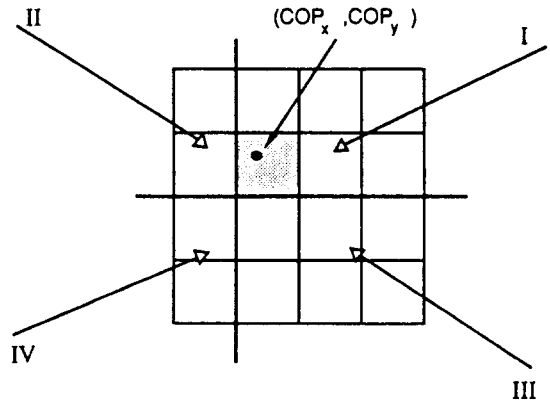

Fig. 10b.

It can be observed that the cases shown in Figs. 9b and 9c can be dealt with by splitting the domain into two or four subdomains respectively, each independent of the others, and able, therefore, to be dealt with as if they were a case of Fig. 9a (Figs. 10a–b).

These subdomains are independent, but because of the division shown in the figure, they are to be processed in the indicated order.

By operating the division of the domain in this way, also in the case of perspective projections, it is possible to consider correct paths as always having as their first element a corner grid unit.

We will now present three possible correct paths for both parallel and perspective projections.

Without changing general cases, the grid unit E_1 may be considered nearest to the line s or to the point (COPx, COPy) in the following examples (cfr. Fig. 11);

Strip path
(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20)
or symmetrically:
(1, 5, 9, 13, 17, 2, 6, 10, 14, 18, 3, 7, 11, 15, 19, 4, 8, 12, 16, 20).

ZigZag path
(1, 2, 5, 3, 6, 9, 4, 7, 10, 13, 8, 11, 14, 17, 12, 15, 18, 16, 19, 20)
or symmetrically:
(1, 5, 2, 9, 6, 3, 13, 10, 7, 4, 17, 14, 11, 8, 18, 15, 12, 19, 16, 20).

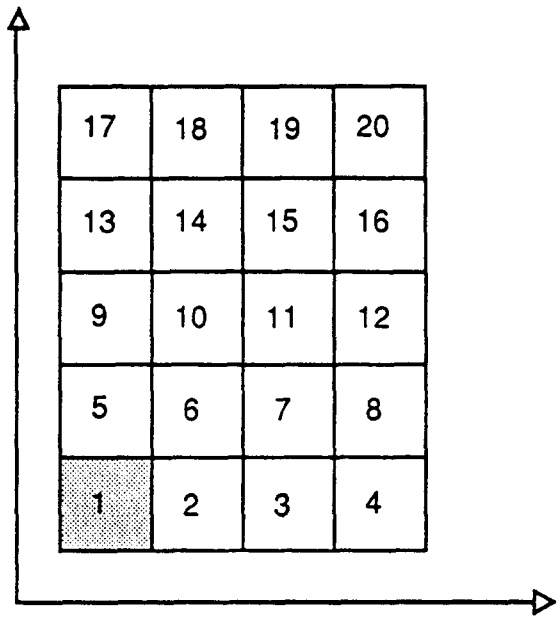

Fig. 11.

Square path:
(1, 2, 5, 6, 3, 7, 9, 10, 11, 4, 8, 12, 13, 14, 15, 16, 17, 18, 19, 20)
or symmetrically:
(1, 5, 2, 6, 9, 10, 3, 7, 11, 13, 14, 15, 4, 8, 12, 16, 17, 18, 19, 20).

5. TAM METHOD FOR CROSS-HATCHED DRAWING OF SINGLE-VALUED CONTINUOUS FUNCTIONS

This method consists in drawing on the screen, in the established order, all the quadrilaterals Q_k, of the vertices:

$$(XS_i, YS_j), (XS_i, YS_{j+1}), (XS_{i+1}, YS_{j+1}), (XS_{i+1}, YS_j),$$

where Q_k are parallel or perspective projections of the $F^*(E_k)$ surface unit.

This method will really only draw the sides of the quadrilaterals; to be precise, any sides common to more than one quadrilaterals will be considered once only. The principle of only drawing the visible sides is to preserve the boundary of the area that is being drawn. This can be done by using two arrays called Masktop and Maskbottom. Any pixels making up the sides of the quadrilaterals will therefore be drawn *if and only if* they lie outside the determined area of invisibility.

Definition. A drawing method can be said to be exact if all and only the visible pixels of the sides of the Q_k appear in the final image.

As has already been suggested in [8], in this paper the idea of the Masktop and Maskbottom arrays will be combined with Bresenham's Drawing Line algorithm, obtaining a very efficient process that can be implemented in hardware.

```
procedure BRESENHAM(XS1, YS1, XS2, YS2: integer)
  procedure UP_PLOT(XS, YS: integer)
  var fd: Boolean
  begin
    fd:=false
    if YS>masktop[XS] then
                          masktop[XS]:=YS
                          fd:=true
    if YS<maskbottom[XS] then
                          maskbottom[XS]:=YS
                          fd:=true
    if fd then PLOT(XS, YS)
  end
begin
  ....
  UP_PLOT(XS, YS)
  ....
end
```

The UP_PLOT procedure presented here is purely indicative: Further arrangements would, in fact, have to be made before it could become operative.

Example

Given the situation presented in Fig. 12a, the quadrilateral of vertices (6, 10), (5, 5), (12, 6), (13, 9) will

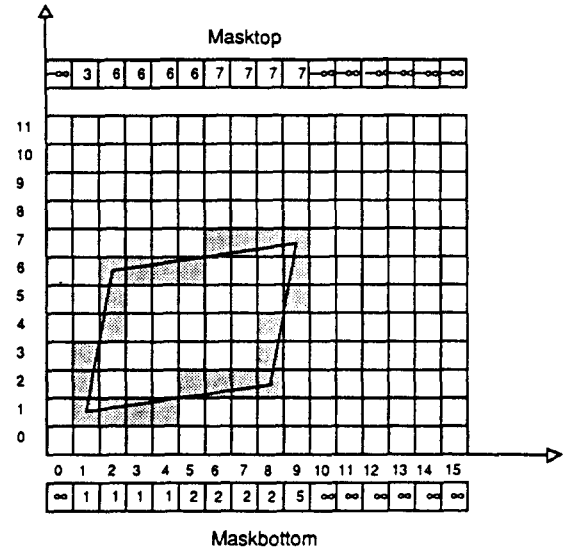

Fig. 12a.

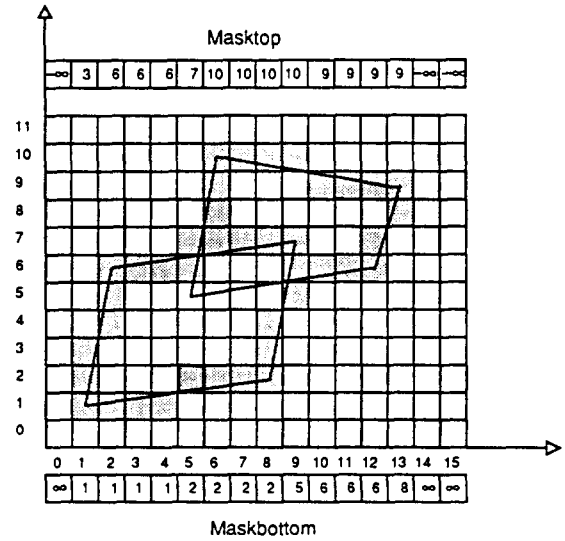

Fig. 12b.

have to be processed in device coordinates. The situation is shown in Fig. 12b after the sides have been drawn in and the two arrays have been updated.

6. THE TAM METHOD IS EXACT FOR PARALLEL PROJECTIONS

Let us consider a plane α perpendicular to the x-y plane and passing through a projector p (Fig. 13a). As has been mentioned, a local ordering will be induced for the $F^*(E_k)$ units intersected; let us take this to be:

$$(F^*(E_{k_i})) \quad i = 1, \ldots, n_l.$$

Let A_1 and B_1 be the intersections of the plane α with the edges of $F^*(E_{k_1})$ and let a_1 and b_1 be the parallel projections of A_1 and B_1 onto the projection plane π (Fig. 13b).

The plane α is projected on a vertical line v; then $F^*(E_{k_1}) \cap \alpha$ points will be mapped on the vertical interval $[a_1, b_1]$ belonging to v.

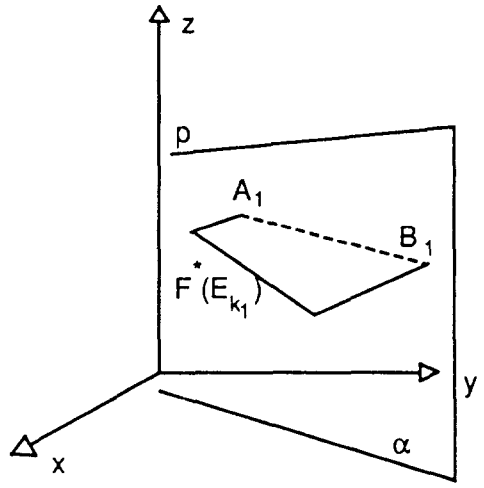

Fig. 13a.

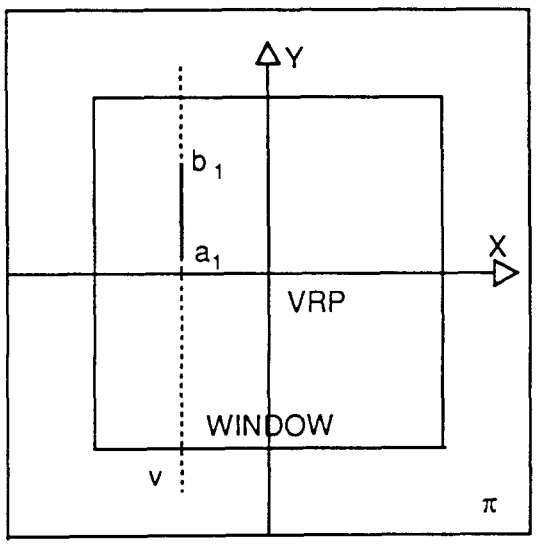

Fig. 13b.

Lemma

Let $(F^*(E_{k_i}))$ $i = 1, \ldots, n_l$ be the local ordering induced for the $F^*(E_k)$ by the plane α.

Suppose that parallel projections map $F^*(E_{k_i}) \cap \alpha$ onto the interval $[a_i, b_i]$ of v. Then

$$\bigcup_{i=1,\ldots,j} [a_i, b_i] = [\min_{i=1,\ldots,j}(a_i), \max_{i=1,\ldots,j}(b_i)] \text{ for each } j \leq n_l$$

Proof

By the continuity of F^*, $F^*(E_{k_1})$ will have a point on α in common with $F^*(E_{k_2})$; therefore the interval $[a_1, b_1]$ will have a value in common with $[a_2, b_2]$ and so:

$$[a_1, b_1] \cup [a_2, b_2] = [\min(a_1, a_2), \max(b_1, b_2)].$$

Supposing that the thesis is true for $j-1$ we will prove it for j. Still on the basis of the continuity of F^*, $F^*(E_{k_j})$ will have a point on α in common with $F^*(E_{k_{j-1}})$; therefore the interval $[a_j, b_j]$ will have a value in common with $[a_{j-1}, b_{j-1}]$; but $[a_{j-1}, b_{j-1}] \subset [\min_{i=1,\ldots,j-1}(a_i), \max_{i=1,\ldots,j-1}(b_i)]$ thus proving the thesis.

This lemma ensures that on each vertical line v, there is a single projected interval (or segment) at each instant i.

Definition. A polygon is said to be vertically convex if its intersection with a vertical line is a single segment.

Similarly a polygon will be defined to be horizontally convex if its intersection with a horizontal line is a single segment.

Definition. We define $\Sigma_{h,q}$ as

$$\Sigma_{h,q} = \bigcup_{i=1,\ldots,q} Q_{k_{h,q}}$$

with $Q_{k_{h,q}}$ as parallel or perspective projection of $F^*(E_{k_{h,q}})$, h as the subscript of a correct path and q as the q-th unit in the path h.

The following will therefore be valid:

Theorem. By considering a correct path h for parallel projections, it follows that $\Sigma_{h,q}$ is a vertically convex region for each $q \leq (NX-1)(NY-1)$.

Proof

Let us suppose, *ab absurdo*, that there is a q such that $\Sigma_{h,q}$ is not vertically convex; it would therefore also follow that the intersection of a vertical line v, with $\Sigma_{h,q}$ would not be a single segment. Yet parallel projections map the plane α onto the line v and, as the path is correct, it will respect the order induced by the plane α; therefore, according to the previous lemma, it is not possible to have more than one mapped segment.

It can therefore immediately be deduced that the TAM method is exact for parallel projections: these generate some $\Sigma_{h,q}$ for each q, which are vertically con-

vex, as has been proved by the theorem; the boundary of a vertically convex polygon, on the other hand, may be accurately described by means of two single arrays (two discrete single-valued functions).

7. THE TAM METHOD IS NOT EXACT FOR PERSPECTIVE PROJECTIONS

By bringing the conditions to mind that were imposed in paragraph 3. for perspective projections, let us consider the sheaf of planes passing through COP and orthogonal to the x-y plane (Fig. 14a).

Let us say that the plane of the sheaf passing through COS is α^* and the angle COS COP T on the plane α^* is Γ. The plane passing through COP and the X-axis of the plane π is β.

Let us then consider that a plane α belonging to the sheaf mentioned is passing through a projector p; Φ is the angle between the planes α and α^* on the plane β.

Therefore, if perspective projection maps the plane α onto the line v of π, this will form an angle Ω with the Y-axis (Fig. 14b) resulting in the following relation:

$$tg(\Omega) = tg(\Phi)/tg(\Gamma) \qquad (3)$$

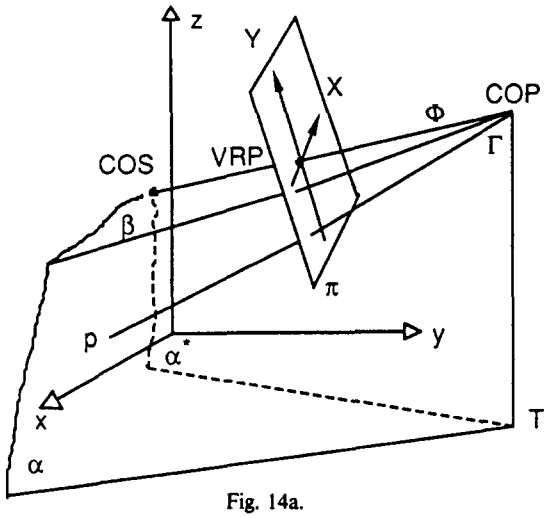

Fig. 14a.

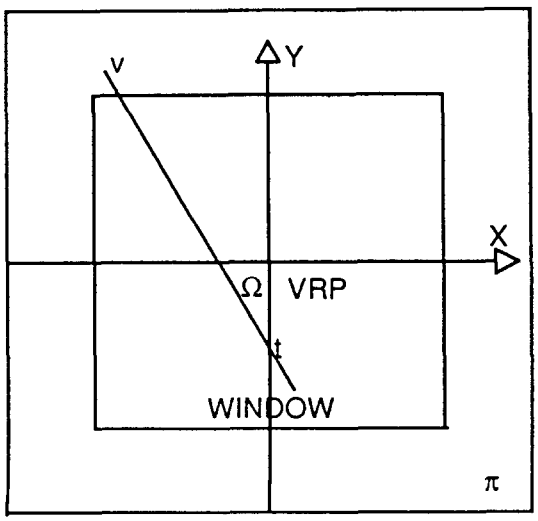

Fig. 14b.

This expression binds the slope of line v to the COP selection and to the plane α of which the line v is projection.

As has already been mentioned such a plane α will induce an ordering among the $F^*(E_k)$ units intersected; let us take this to be:

$$(F^*(E_{k_i})) \quad \text{for} \quad i = 1, \ldots, n_l$$

Let A_1 and B_1 be the intersections of the plane α with the edges of the $F^*(E_{k_1})$ unit and let a_1 and b_1 be the perspective projection of A_1 and B_1 upon v.

Taking t, projection of T, to be the origin on the line v, then $F^*(E_{k_1}) \cap \alpha$ will be mapped on the real interval $[a_1, b_1]$.

Just as was the case with parallel projections, in perspective projections the intervals $[a_i, b_i]$ for $i = 1, \ldots, n_l$ are never separated on line v.

Given (3), the v lines are not usually vertical, which would lead us to consider that for vertical lines there is no guaranteed "continuity" of intervals $[a_i, b_i]$ for $i = 1, \ldots, n_l$, a necessary prerequisite for the precision of the TAM method.

This means, therefore, that an $F(x, y)$, a COP and a correct path can always be determined, so that there is at least one $\Sigma_{h,q}$ which is not vertically convex and which would result in an inexact drawing.

Example

Let the following grid of values be assigned:

$$(x_i, y_j, F(x_i, y_j)) \quad i = 1, \ldots, 7, \quad j = 1, \ldots, 5$$

with

$$x_i = i - 4$$

$$y_j = j - 3$$

$$F(x_i, y_j) = \begin{cases} 10 & \text{for } i = 3, \quad j = 3 \\ 0 & \text{otherwise.} \end{cases}$$

If the Strip path is used and the surface assigned is observed from COP $\equiv (1, -7, 10)$, the result obtained is shown in Figs. 15a–b.

It has therefore been proved that the TAM method is inexact for perspective projections; the boundary of a vertically non-convex polygon may not, in fact, be accurately described by means of two single arrays (two discrete single-valued functions).

8. THE TAM METHOD IS EXACT FOR PERSPECTIVE PROJECTIONS WITH TWO VANISHING POINTS FOR x AND y AXES

As has been shown in the preceding paragraph, it can be deduced that the TAM method is exact for perspective projections only when $\Omega = 0$. By examining (3), it can be seen that Ω is zero only when $\Phi = 0$ or $\Gamma = \pi/2$. In the former case we are forced back to parallel projections; in fact, none of the α planes containing p projectors ever cut the plane α^*. In the latter

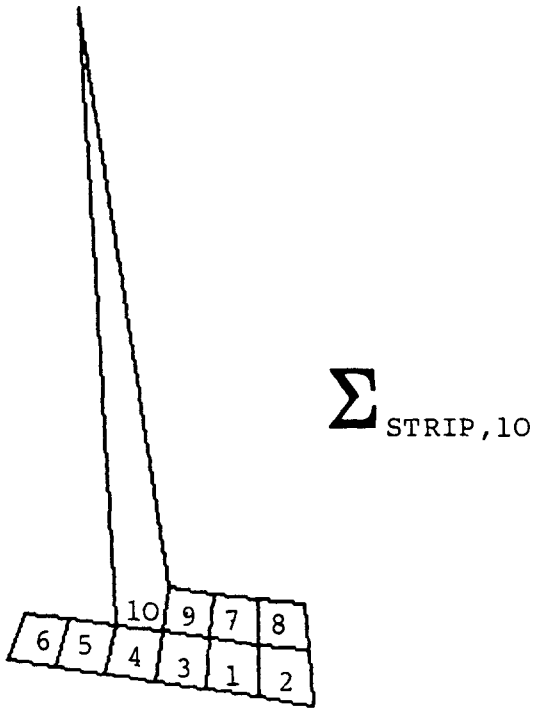

Fig. 15a.

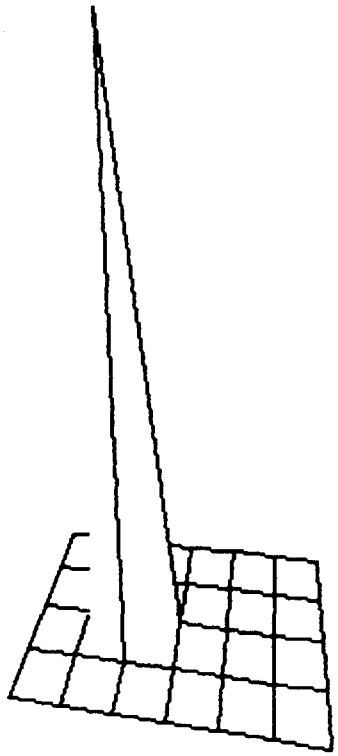

Fig. 15b.

case the plane π is orthogonal to the x-y plane and therefore COP$z \equiv$ COSz.

It can also be said that there is a direct relation between the width of angle Ω and the accuracy of the method; in fact if Ω approaches zero, then the method improves in precision and by (3) Φ approaches zero and/or Γ approaches $\pi/2$. Summing up, this means that the TAM method is exact for perspective projections when a COP is chosen whereby COP$z \equiv$ COSz; i.e., provided that the perspective projections have two vanishing points for x and y axes.

The scheme described in paragraph 3. may therefore be modified by altering COSz until it is the same as the z-coordinate of the COP selected. This will guarantee the TAM method to be exact.

Yet if, as in paragraph 3., a squared window containing the S sphere projection is defined upon π, the projected images will be lengthened along the Y-axis as COP$z \gg \max(F(x_i, y_j))$ or COP$z \gg \min(F(x_i, y_j))$, thus giving a real, yet unpleasant effect.

As an alternative, choosing the window as the smallest rectangle containing the S projection, will provide a more pleasant image similar to the one obtained by following the instructions in paragraph 3.

However, by modifying the scheme in this way, COP selection is restricted; in fact, in order to guarantee that the projection exists on every point of the function, (COPx, COPy) will always have to be outside the $F(x, y)$ domain.

9. EXPERIMENTAL SUGGESTIONS

In preceding paragraphs we have seen that it is not possible to forecast the accuracy of the TAM method for perspective projections with three vanishing points. As soon as a $\Sigma_{h,q}$ is present which is not vertically convex, instead of describing its boundary, the two array mask being used will, in fact, describe the boundary of the smallest vertically convex polygon containing $\Sigma_{h,q}$, which we are going to call $V(\Sigma_{h,q})$ (Fig. 16).

If, therefore, one pixel of a side of $Q_{h,k_{q+1}}$ falls in the difference region:

$$V(\Sigma_{h,q}) - \Sigma_{h,q}$$

it will be treated as if it were invisible, although it is outside the $\Sigma_{h,q}$ region.

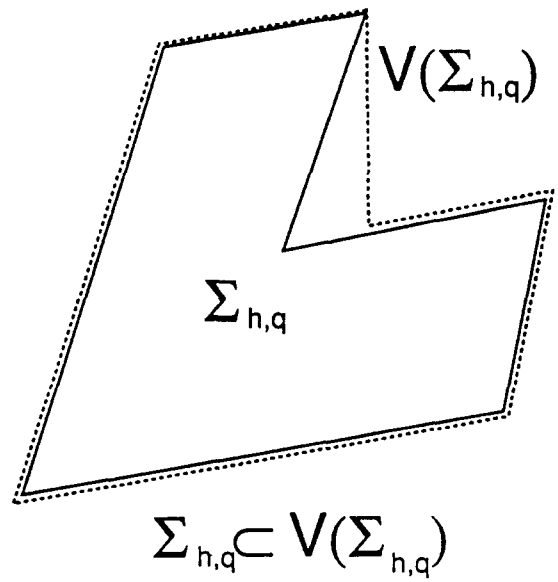

Fig. 16.

It can therefore be observed that the TAM method "is not exact by default" for perspective projections; in other words, it does not draw visible pixels either, by contrast with methods which are "not exact by excess" yet which also draw invisible pixels.

However, the use of the TAM method will still remain a valid alternative, as the problems arising occur mainly where the data of a function vary rapidly or where COP near to the surface are given, while the method is characterized by its considerable execution speed.

So if we wish to continue using the TAM method it is best to analyze whether, with the tools available, it would be possible to reduce the difference region for each q.

In paragraph 4. it was seen that once a COP and the surface domain had been set, there was more than one correct path. Obviously different $\Sigma_{h,q}$ areas and consequently difference regions of varying widths for each q will be induced by each path.

There is experimental evidence that the Strip path is the most reliable of those suggested (Figs. 17a–c).

Whichever path is selected, it may either be developed along the x-axis or the y-axis; preference would instinctively be given, and this has been confirmed by experimental evidence, to the more horizontal of the two axes, once projected onto the plane π.

It would finally be advisable to introduce a rule defining the order and direction that the sides of each Q_k are to be drawn in. Let us suppose that the x-axis, after projection, is the more horizontal of the two and that the first grid unit is E_1, then the sides of Q_k may be drawn as (Fig. 18):

a) $(XS_i, YS_{j+1}) \dashrightarrow (XS_i, YS_j)$

b) $(XS_i, YS_j) \dashrightarrow (XS_{i+1}, YS_j)$

c) $(XS_{i+1}, YS_j) \dashrightarrow (XS_{i+1}, YS_{j+1})$

d) $(XS_{i+1}, YS_{j+1}) \dashrightarrow (XS_i, YS_{j+1})$

where once a segment is considered as the side of a previous Q_k it does not need to be reexamined.

Example

Let us consider the case in Fig. 19: Let us say that E_5 is the grid unit containing (COPx, COPy) or the nearest to it. The domain will therefore still be divided into four subdomains by lines $x = x_2$ and $y = y_2$. Let us suppose that for each subdomain the Strip path has been chosen along the x-axis. In order to minimize the difference region $V(\Sigma_{h,q}) - \Sigma_{h,q}$ for each q, instead of examining one subdomain at a time, we would advise one strip being examined at a time for each subdomain; in other words we would recommend the following correct path overall:

$$(E_5, E_6, E_4, E_2, E_3, E_1, E_8, E_9, E_7)$$

Finally, by following the afore-mentioned rule, the Q_k sides will be drawn in the following order and direction, where the subscript pairs of the endpoints are indicated:

Q_5:	(2, 3) - -> (2, 2);	(2, 2) - -> (3, 2);	(3, 2) - -> (3, 3);	(3, 3) - -> (2, 3);
Q_6:	---	(3, 2) - -> (4, 2);	(4, 2) - -> (4, 3);	(4, 3) - -> (3, 3);
Q_4:	---	(2, 2) - -> (1, 2);	(1, 2) - -> (1, 3);	(1, 3) - -> (2, 3);
Q_2:	(2, 1) - -> (2, 2);	---	(3, 2) - -> (3, 1);	(3, 1) - -> (2, 1);
Q_3:	---	---	(4, 2) - -> (4, 1);	(4, 1) - -> (3, 1);
Q_1:	---	---	(1, 2) - -> (1, 1);	(1, 1) - -> (2, 1);
Q_8:	(2, 4) - -> (2, 3);	---	(3, 3) - -> (3, 4);	(3, 4) - -> (2, 4);
Q_9:	---	---	(4, 3) - -> (4, 4);	(4, 4) - -> (3, 4);
Q_7:	---	---	(1, 3) - -> (1, 4);	(1, 4) - -> (2, 4);

10. THE FOUR ARRAY MASK METHOD

Having worked out in previous paragraphs why the TAM method is not exact, we are now going to suggest a variant called the Four Array Mask (FAM) method which considerably improves final representation and at the same time ensures a comparable speed of execution.

The method under examination sets out to reduce the extent of the difference region $V(\Sigma_{h,q}) - \Sigma_{h,q}$ by introducing a new pair of arrays called Maskleft and Maskright. The two new arrays perform similar tasks along the horizontal axis as Masktop and Maskbottom do along the vertical axis; so if the latter ensure drawing precision in the case of vertically convex $\Sigma_{h,q}$, the addition of the new pair of arrays even provides an exact result for horizontally convex $\Sigma_{h,q}$.

Obviously, as perspective projections cannot guarantee that the $\Sigma_{h,q}$ are vertically convex, neither can they guarantee that they are horizontally convex; on the other hand, the difference region is considerably reduced. In fact if by analogy we define $O(\Sigma_{h,q})$ as the smallest horizontally convex polygon containing $\Sigma_{h,q}$ (Fig. 20), then we have as difference region:

$$V(\Sigma_{h,q}) \cap O(\Sigma_{h,q}) - \Sigma_{h,q}$$

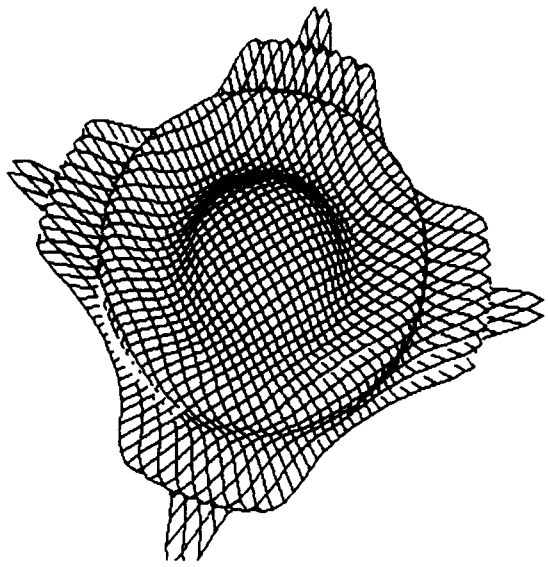

Fig. 17a. $F(x, y) = 4\cos((x^2 + y^2)/10$ with $(x, y) \in [-10, 10] \times [-10, 10]$ and $NX = NY = 40$ observed from COP (3, 5, 20). Output obtained using Strip Path.

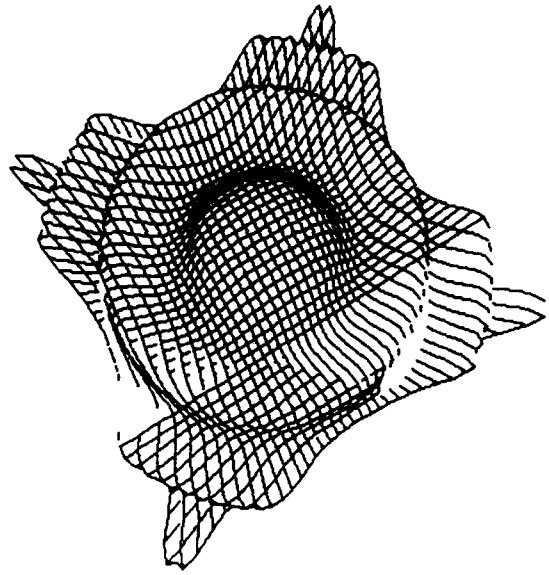

Fig. 17c. $F(x, y) = 4\cos((x^2 + y^2)/10$ with $(x, y) \in [-10, 10] \times [-10, 10]$ and $NX = NY = 40$ observed from COP (3, 5, 20). Output obtained using Square Path.

where, obviously

$$V(\Sigma_{h,q}) \cap O(\Sigma_{h,q}) \subset V(\Sigma_{h,q}).$$

Some experimental results are shown in the following: Figs. 21a–b show the results obtained with the TAM and FAM methods applied to the linear surface patches worked out from the following set of values:

$$(x_i, y_j, F(x_i, y_j)) \quad i = 1, \ldots, 20, \quad j = 1, \ldots, 20$$

with

$$x_i = 1 + (i - 1)/2$$

$$y_j = 1 + (j - 1)/2$$

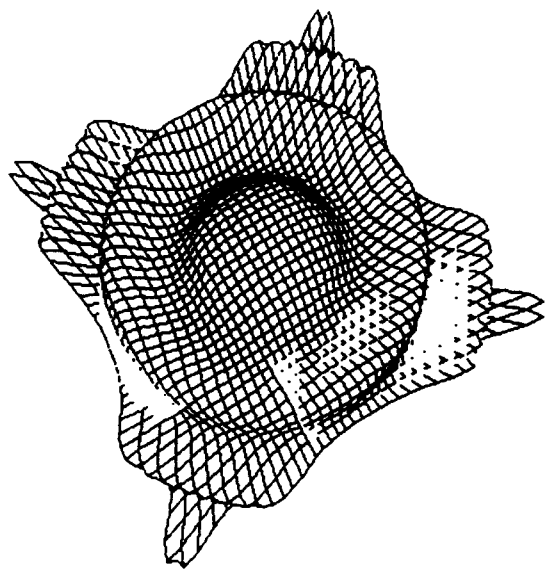

Fig. 17b. $F(x, y) = 4\cos((x^2 + y^2)/10)$ with $(x, y) \in [-10, 10] \times [-10, 10]$ and $NX = NY = 40$ observed from COP (3, 5, 20). Output obtained using ZigZag Path.

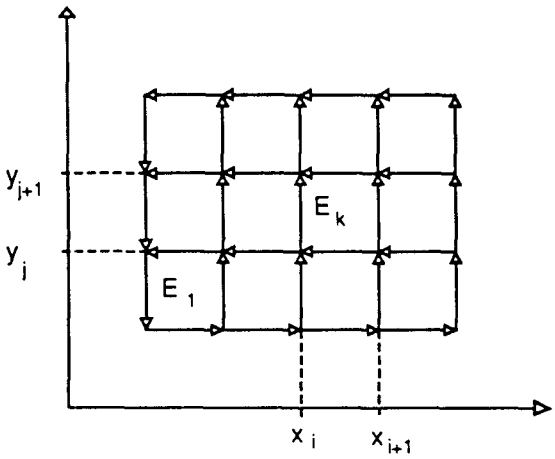

Fig. 18.

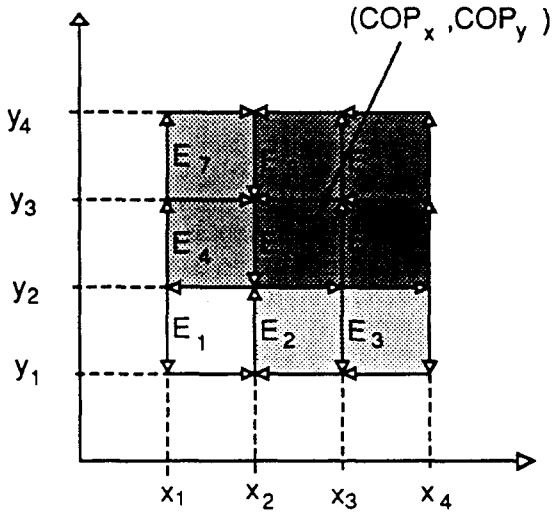

Fig. 19.

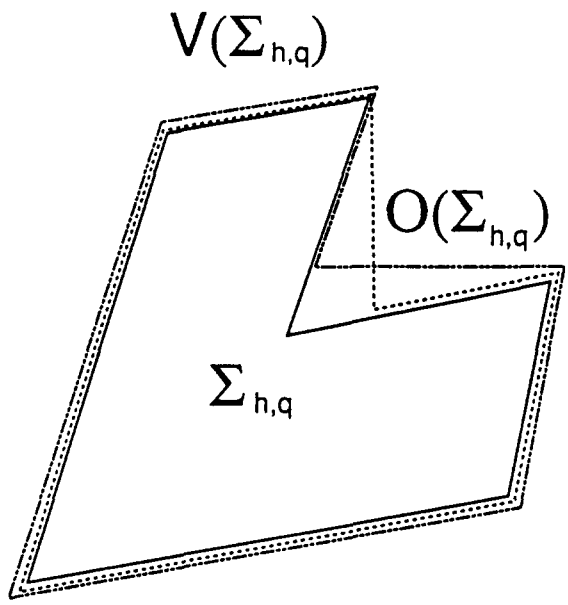

Fig. 20.

$$F(x_i, y_j) = 10 \quad \text{for} \quad i = 3, 4, \quad j = 3, \ldots, 18$$

$$0 \quad \text{otherwise}$$

observed from the COP $\equiv (-5, 12, 18)$;

Figs. 22a–b show the results obtained with the two methods applied to the surface:

$$F(x, y) = 4\text{sen}((x^2 + y^2)/10) \quad \text{with}$$

$$(x, y) \in [-6, 6]x[-6, 6] \quad \text{and} \quad NX = NY = 40,$$

observed from the COP $\equiv (10, 5, 10)$.

Finally Table 1 gives a comparison of the performance of the two methods with the Strip path setting. The results of the tests shown here were obtained by using a DEC GPX II graphic workstation with a VMS 4.4 operating system; the programs were developed in Pascal.

Note that symmetry is induced by the additional pair of arrays in terms of the X and Y axes. Consequently, when selecting a path, there is no longer any point in giving preference to development along one axis rather than another; the order and direction that the Q_k sides are drawn in are no longer of any relevance.

As a consequence of the above, this variant does not require the use of an algorithm conceived to draw directed segments (Bresenham from the first to the second endpoint). In particular our realization uses an accelerated drawing line algorithm exploiting the bidirectional concept[10].

One way of dealing with the four arrays mask is outlined below:

```
procedure BDR_BRESENHAM(XS1, YS1, XS2, YS2:
integer)
  procedure ORIZ_UP_PLOT(XS, YS: integer)
  var fd: Boolean
  begin
    fd:=false
    if YS>masktop[XS] then
                        masktop[XS]:=YS
                        fd:=true
    if YS<maskbottom[YS] then
                          maskbottom[XS]:=YS
                          fd:=true
    if XS>maskright[YS] then
                        fd:=true
    if XS<maskleft[YS] then
                        fd:=true
    if fd then PLOT(XS, YS)
  end
  procedure ORIZ_UP(XA, XB, YS:integer)
  begin
    if XA<maskleft[YS] then
                        left[YS]:=XA
    if XB>maskright[YS] then
                         right[YS]:=XB
  end
```

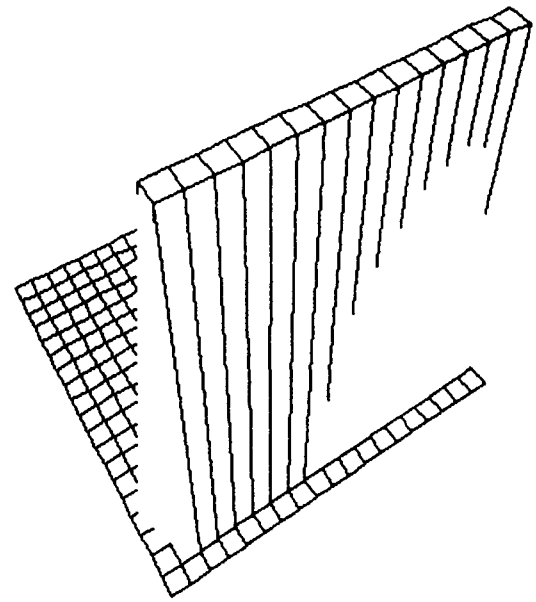

Fig. 21a. Output obtained using TAM method.

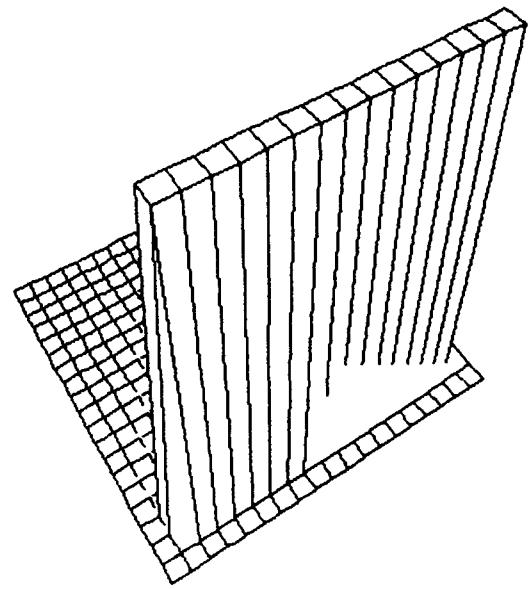

Fig. 21b. Output obtained using FAM method.

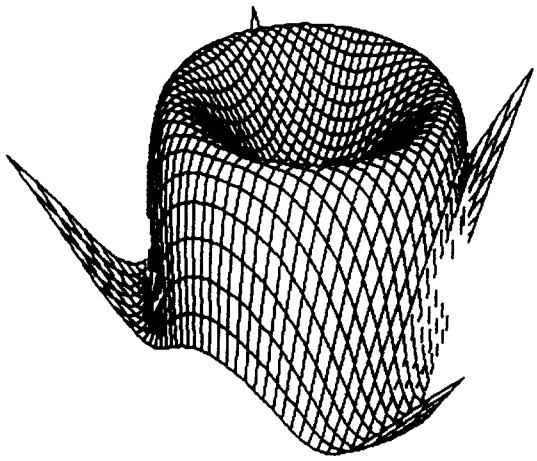

Fig. 22a. Output obtained using TAM method.

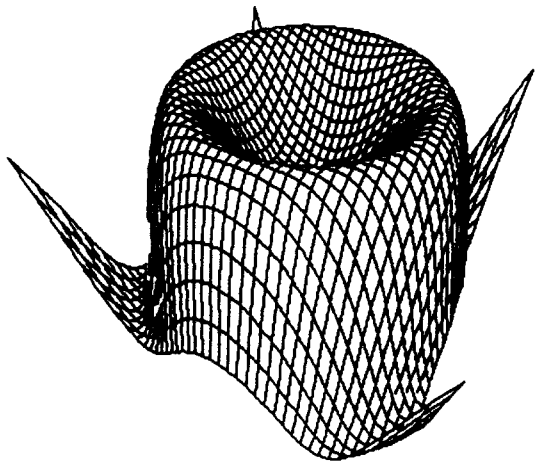

Fig. 22b. Output obtained using FAM method.

```
procedure VERT_UP_PLOT(XS, YS:integer)
var fd:Boolean
begin
  fd:=false
  if YS>masktop[XS] then
                          fd:=true
  if YS<maskbottom[XS] then
                             fd:=true
  if XS>maskright[YS] then
                           maskright[YS]:=XS
                           fd:=true
  if XS<maskleft[YS] then
                          maskleft[YS]:=XS
                          fd:=true
  if fd then PLOT(XS, YS)
end
procedure VERT_UP(XS, YA, YB:integer)
begin
  if YA<maskbottom[XS] then
                             maskbottom[XS]:=YA
  if YB>masktop[XS] then
                          masktop[XS]:=YB
end
begin
  ....
  if dy<=dx then .....
                 ORIZ_UP_PLOT(XS, YS)
                 .....
                 ORIZ_UP(XA, XB, YS)
                 .....
               else .....
                 VERT_UP_PLOT(XS, YS)
                 .....
                 VERT_UP(XS, YA, YB)
                 .....
  ....
end
```

11. EXACT METHODS OF DRAWING FOR PERSPECTIVE PROJECTIONS

It has been shown why the TAM and FAM methods are not usually exact for perspective projections. It has also been shown that, for single-valued continuous functions, $F^*(E_k)$ sorting can be worked out with low computational complexity. Starting with the latter results, further alternative methods to the mask ones can be worked out, which are exact for perspective projections although they are not so rapid. Subsequently two of these methods will be referred to, which, although slow in most other cases do prove here to be most competitive under these conditions; in fact they do not require $F^*(E_k)$ sorting, which, as is known, is the most expensive phase of the methods of elimination of hidden lines or surfaces.

Newell, Newell and Sancha algorithm

This is the idea of developing an order of priority for all the quadrilaterals in the set, according to their depth. Once this sorting has been determined, the quadrilaterals are scan-converted into the frame buffer, one at a time, starting with the deepest.

In our case a correct sorting is immediately given by one of the paths previously examined. In order to exploit it, we have just to scan the grid units in the list, from the last one up to the first.

Visibility buffer algorithm

This is the idea of proceeding as per the TAM method, but using a bidimensional array of Boolean (visibility buffer) of the dimensions of the screen so as to accurately preserve the $\Sigma_{h,q}$ regions as well as their boundaries, by conveniently setting some elements of the array.

The quadrilaterals are examined one at a time, according to whichever path has been worked out as cor-

Table 1. $F(x, y) = 4 \text{ sen } ((x^2 + y^2)/10$ with $(x, y) \in [-6, 6] \times [-6, 6]$ and $NX = NY = 40$.

	TAM method		FAM method	
COP	Pixel %	Time (sec)	Pixel %	Time(sec)
(10,10,10)	100	9.32	100	10.16
(10,5,10)	93.9	8.05	99.1	9.61
(10,0,10)	92.9	8.02	97.5	9.40
(4,4,15)	97.7	7.70	100	8.98
(4,4,-15)	95.0	9.91	100	11.59
(4,0,20)	97.9	8.87	100	10.04

rect (ranging from the nearest to the furthest). Each quadrilateral is examined as follows:

1. The sides are considered pixel by pixel; visibility is evaluated according to whether or not they are outside the region that has been setted to true in the buffer;
2. The quadrilateral is scan-converted into the buffer by setting the correspondent elements of the bidimensional array to true.

12. CONCLUSIONS

In this paper the TAM method has been reexamined as a method of eliminating hidden surfaces by showing, and clearly separating the sorting phase from the drawing phase of the surface units.

A general theory has also been presented for working out the correct path for both parallel and perspective projections, and it has been shown that the TAM method is exact for the former but not for the latter projections, with the exception of projections with two vanishing points for x and y axes.

The FAM method was generally suggested for perspective projections; this was theoretically shown to be more accurate, and experimental evidence showed that execution time was comparable with that of the TAM method.

Generally speaking, however, exact representation can be found in different drawing methods; on account of their simplicity and efficiency the Newell, Newell and Sancha and visibility buffer methods were considered.

Full presentation and detailed analysis of the realization of the methods examined may be found in [11].

REFERENCES

1. B. Kubert, J. Szabo and S. Giulieri, The perspective representation of function of two variables. *Journal of ACM* **15** (2), 193–204 (1968).
2. H. Williamson, Hidden-line plotting program [J6]. *Comm. ACM* **15** (2), 100–103 (1972).
3. T. J. Wright, A two-space solution to the hidden line problem for plotting function of two variables. *IEEE Trans. on Comp.* **c-22** (1), 28–33 (1973).
4. S. L. Watkins, Masked three-dimensional plot program with rotations [J6]. *Comm. ACM* **17** (9), 520–523 (1974).
5. J. Butland, Surface drawing made simple. *CAD* **11** (1), 19–22 (1979).
6. C. Cavagna, F. Maggi and F. Panin, Rappresentazione prospettica con eliminazione delle linee nascoste di superficie definite da funzioni di due variabili. *Pixel* **2**, 5–15 (1982).
7. W. T. C. Sowerbutts, A surface-plotting program suitable for microcomputers. *CAD* **15** (6), 324–327 (1983).
8. V. Skala, An interesting modification to the Bresenham algorithm for hidden-line solution. *NATO ASI Series* **F17**. In: R. A. Earnshaw (Ed.), *Fundamental Algorithm for Computer Graphics*, Springer-Verlag, New York (1985) and *Comp. Graphics Forum* **6** (4), 343–347 (1987).
9. E. Sutherland, R. F. Sproull and R. A. Shumaker, A characterization of ten hidden-surface algorithms. *Comp. Surveys* **6** (1), 1–55 (1974).
10. G. Casciola, Basic concepts to accelerate line algorithm. *Comp. & Graphics* **12**(3/4), 489–502 (1988).
11. L. Alvisi and G. Casciola, Two and four array mask algorithms in practice. Technical Report, Department of Mathematics, University of Bologna, Italy (1988).
12. J. D. Foley and A. Van Dam, *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, New York (1982).