



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) Publication number:

0 411 739 A2

(12)

EUROPEAN PATENT APPLICATION

(21) Application number: **90302573.2**

(51) Int. Cl.⁵: **G09G 5/28**

(22) Date of filing: **09.03.90**

(30) Priority: **01.08.89 US 388339**

(43) Date of publication of application:
06.02.91 Bulletin 91/06

(84) Designated Contracting States:
DE FR GB IT NL SE

(71) Applicant: **ADOBE SYSTEMS INC.**
1585 Charleston Road
Mountain View California 94039-7900(US)

(72) Inventor: **Paxton, William H.**
25660 La Lanne Court
Los Altos Hills, California 94022(US)
Inventor: **Schiller, Stephen N.**
507 6th Avenue
Menlo Park, California 94025(US)

(74) Representative: **Wombwell, Francis et al**
Potts, Kerr & Co. 15, Hamilton Square
Birkenhead Merseyside L41 6BR(GB)

(54) **Method for editing character bitmaps at small sizes.**

(57) A method to detect and fix certain topological errors in character bitmaps by (1) enumerating the pixels in an order determined by the path topology, (2) searching for sequences of pixels corresponding to a pointed feature in the character that touches other parts of the character bitmap, and (3) editing the bitmap to fix such incorrect contacts.

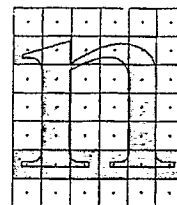


FIGURE 1A



FIGURE 1D

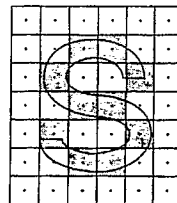


FIGURE 1B

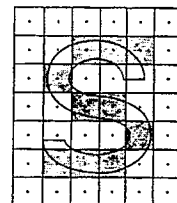


FIGURE 1E

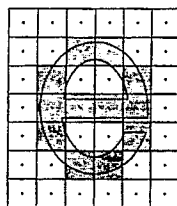


FIGURE 1C

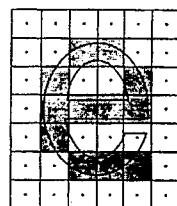


FIGURE 1F

EP 0 411 739 A2

METHOD FOR EDITING CHARACTER BITMAPS AT SMALL SIZES

FIELD OF THE INVENTION

In modern computer systems, it is often desirable to print or display characters in various sizes on paper, film or a computer screen. When the size of the character is large relative to the resolution of the display or print device, it is relatively easy to choose which picture elements or pixels should be printed or displayed in order to make a readable character. However, when the size of the character is small in relation to the resolution of the display, it is much more difficult to choose which pixels to display in order to make the character as distinct and recognizable as possible. The current invention relates to an improved method of legibly displaying characters at low resolution.

BACKGROUND OF THE INVENTION

Traditionally, characters have been printed using metal type which allows very detailed rendering of a character, including subtle curves and very fine lines. In modern computer devices, characters are defined on raster devices such as video display terminals or by using a multi-pin print head. Characters can be printed on a surface or displayed on a video screen as a series of dots which are printed or turned on in order to approximate as closely as possible the ideal shape of the character. When characters are small enough relative to the resolution of the display device, choosing which pixels should be displayed to accurately represent the character becomes more complex than when the character is large. A typical video monitor can display about 72 pixels per inch. At this resolution it is difficult to display legibly most type faces smaller than about twenty pixels tall.

An ideal representation of the character is usually defined in "character space" at very high resolution as one or more areas bounded by an outline or path. A character consists of one or more continuous black areas. For instance the letter "O" consists of a single closed loop, the letter "d" consists of a loop connected to a line and the letter "i" consists essentially of a dot a short distance away from a line which may have additional details such as serifs. One way of describing a character involves defining an outline of the outer edge of each contiguous black portion of the character and then filling that outline to display the character. Since characters are usually printed in dark ink on a light background, one can describe filled areas as black but one skilled in the art will recognize that characters which are light on a dark background,

commonly used in video displays, are also within the teachings of this invention. This path can be represented as a sequential series of curves and/or linear line segments called edges. If a black area has interior white areas as, for instance, in the letter "O", each interior white area can also be defined by a path consisting of a series of edges.

When tracing or displaying such a character, it is generally useful to trace the edges in a consistent direction, either clockwise or counter-clockwise. If edges of an outside path are traced in the counter-clockwise direction, then the area to the left of that edge will always be black and the area to the right will always be white. If the path is traced in the clockwise direction, the black area will be on the right of the edge. Enclosed white areas should be traced in the direction opposite to the exterior path so that the black area is on the same relative side of the edge.

When a character is displayed on a raster device, those pixels which fall within the black area of the character should be displayed, that is, they should be printed on a surface or turned on for a video display. Various methods of filling a character outline are known to those skilled in the art. At high resolution or when the character is very large, multiple pixels may fall within each black area and the character can be displayed in great detail. When the character is reduced to a small size, however, or the resolution of the device is limited, certain black areas may no longer cover multiple pixels and in fact may cover only a fraction of a pixel. Displaying small characters on a device of limited resolution has been a persistent problem in the past. This is illustrated in Figure 1 by a character on an 6x7 matrix. In Figure 1 the outlines of the characters "n", "s" and "e" are illustrated. The raster display can only turn on or off entire pixels as represented by the elements of the 6x7 matrix.

One prior approach to this problem is the center point fill method and the improved fill method described in the copending application entitled "Dropout-Free Center Point Fill Method For Displaying Characters" by the same inventors and filed on the same date as the present application. One problem that occurs with many fill methods is that certain character features may be found in close proximity. As the display resolution decreases the pixel fill method chosen sometimes turns on pixels that cause parts of the character bitmap to touch each other that actually should not be in contact. This introduces errors in the topology of the character that greatly reduce the legibility. For example, in Figures 1A, 1B and 1C the "n"

closes in at the bottom instead of having an open space between the legs, and the curves of the "e" and the "s" both touch the main body of the character and change the bitmap into something like a small "8".

One object of this invention is to turn on pixels according to the outline of a character using one of the fill rules such as center point fills, and modify the display by turning off or moving pixels that cause the character to close improperly.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates pixel mapping of characters using a simple or modified center point fill method without using (1A, 1B, 1C) and using (1D, 1E, 1F) the method of the present invention.

Figure 2 illustrates certain configurations of groups of four pixels and the corresponding path enumeration.

Figure 3 illustrates groups of four pixels in which the start pixel and the end enumerated pixel are identical.

Figure 4 illustrates certain configurations of groups of three pixels and the corresponding path enumeration.

Figure 5 illustrates groups of three pixels in which the start and end enumerated pixel are identical.

Figure 6 illustrates groups of three pixels in which the end pixel touches corner-to-corner with the start pixel.

Figures 7, 8 and 9 illustrate tests for conflict which are part of this invention.

Figure 10 illustrates resolution of a "move" or "delete" decision.

Figure 11 illustrates an additional pixel pattern.

SUMMARY OF THE INVENTION

The present invention details a method to detect and fix certain topological errors in character bitmaps by (1) enumerating the pixels in an order determined by the path topology, (2) searching for sequences of pixels corresponding to a pointed feature in the character that touches other parts of the character bitmap, and (3) editing the bitmap to fix such incorrect contacts.

DETAILED DESCRIPTION OF THE INVENTION

The concept of describing characters by means of an outline has been explained above. The path of the outline can be traversed in either a clockwise or counter-clockwise direction, and if the path is traversed in a counter-clockwise direction,

the black area of the character will be to the left of the path and the background or non-character area will be to the right of the path. The method of this invention can be used by tracing the path in either direction, but a counter-clockwise traverse will be assumed for the purpose of these examples. The path is thus assumed to be oriented so that the inside of the character is on the left as you face in the direction of the path. If the character is outlined by more than one path, each such subpath is enumerated separately. Enumeration of a character outline is simply a listing of each displayed pixel which includes or is adjacent to each portion of the character path.

Enumerating the Pixels Along the Edge of the Character

The method of enumeration should preferably match the fill method that was used to turn on pixels in the first place. Assuming a center point fill method with modifications to prevent dropout (as described in the co-pending patent application referenced supra), output or include a pixel in the enumeration whenever the path crosses a horizontal or vertical midline connecting pixel centers. Within a pixel there are four midlines to consider, vertical midlines above and below the center point, and horizontal midlines left and right of the center point. For each of these four midlines there are two cases depending on the orientation of the path when it crosses the midline. Trace the path of the character outlines and do the following tests to decide which pixel to output when the path crosses one of the four midlines. The "current pixel" means the one containing the midline that has been crossed.

If the path intersects the center of the current pixel, then output the current pixel.

If the path crosses the left horizontal midline from top to bottom, or crosses the right horizontal midline from bottom to top, or crosses the top vertical midline from right to left, or crossing the bottom vertical midline from left to right, then output the current pixel.

If the path crosses the left horizontal midline from bottom to top, then output the current pixel if it is turned on in the bitmap, else output the pixel to its left.

If the path crosses the right horizontal midline from top to bottom, then output the current pixel if it is turned on in the bitmap, else output the pixel to its right.

If the path crosses the top vertical midline from left to right, then output the current pixel if it is turned on in the bitmap, else output the pixel above.

If the path crosses the top vertical midline from the right, then output the current pixel if it is turned on the bitmap, else output the pixel below.

A path section that crosses a vertical midline exactly at the boundary between two pixels is considered to be in the bottom pixel if it is oriented right to left and in the top pixel otherwise.

A path section that crosses a horizontal midline exactly at the boundary between two pixels is considered to be in the right pixel if it is oriented top to bottom and in the left pixel otherwise.

If two or more sequential crossings cause the same pixel to be output, discard all but one so that no pixel occurs immediately following itself in the enumeration. This simplifies the pattern matching in the next stage. Copy the first three pixels to the end of the enumeration for the path so that the pattern matching does not have to worry about checking patterns that "wrap around" the place where the path begins and ends. The result of the enumeration part of the method of this invention is an enumeration or list of pixels in the order by which a path traverses the pixels.

Matching

The matching process goes through the list of pixels output in the enumeration stage to identify sequences corresponding to a pointed feature in the character that touches other parts of the character bitmap. Examples of pointed features are diagrammatically shown in Figs. 2-6. When such a contact is discovered, one or more edits are recorded for the final stage.

The patterns corresponding to pointed character features or pointed groups of pixels consist of three or four pixels arranged in a counterclockwise sequence. The sequential ordering of the pixels is indicated by the numbers inside the cells in the figures 2-11. Their relative positions in the display are as shown. (Since the path is oriented with the inside of the character on the left, a series of pixels forming an outward directed point will create a counterclockwise pattern.) The four-pixel patterns (Figures 2, 3) have a two pixel point (two point pixels), and the three-pixel patterns (Figures 4, 5) have a one pixel point a point pixel. All possible matches are tested. For example, a sequence of pixels may be part of both a four-pixel pattern and a three-pixel pattern and edits may be recorded for each match.

Once a point pattern has been found, the match process checks to see if any point pixels touch other pixels in the character bitmap that they should not touch. Pixels may "touch" other pixels by intersecting side-to-side (top-to-bottom is equivalent) or intersecting at a corner. In Figure 2A, for

example, pixel 1 touches pixels 2 and 4 side-to-side and touches pixel 3 corner-to-corner. In the case of a four-pixel pattern as in Figures 2 or 3 test for pixels 2 or 3 touching a black pixel in the direction away from pixels 1 and 4. Both side-to-side touches and corner-to-corner touches were considered. For example, in the configuration of Figure 7, pixels **B**, **C** and **D** touched pixel 2 in the direction away from pixels 1 and 4 and pixels **A**, **B** and **C** touched pixel 3 in the direction away from pixel 4. If any of **A**, **B**, **C** or **D** were black, then an incorrect touch existed.

If only one of pixels 2 or 3 touched a black pixel, that pixel was marked for deletion and the other was left. Thus in Figure 7 if only **A** was black, then 3 was marked for deletion and 2 was left as is. If only **D** was black then 2 was marked and 3 was left.

If both 2 and 3 touched one or more black pixels, then one was deleted and the other was either deleted or moved.

"Moving" a pixel means turning off the current pixel and turning on an adjacent one.

If 2 was side-to-side with 4, then 2 was marked for deletion and 3 was considered for moving beside 4. (See Figures 7, 2B, 2E, 2I, 2L)

If 3 was side-to-side with 1, then 3 was marked for deletion and 2 was considered for moving beside 1. See Figures 2C, 2F, 2H, 2K.

If 2 was side-to-side with 1 and 3 was side-to-side with 4, then both were marked for deletion. See Figures 2A, 2D, 2G, 2J.

In the case of a three-pixel pattern where the end pixel is side-to-side with the start pixel (i.e., pixels 1 and 3 in Fig. 4) test for pixel 2 touching a black pixel in the direction away from pixels 1 and 3. For example, in the configuration of Figure 8, if any of **b**, **B** or **C** are black, then an incorrect touch exists and pixel 2 is marked for deletion.

The same test is made in the case of a three-pixel pattern that ends at the same pixel as it starts if the pattern is horizontal or vertical rather than diagonal. (Figure 5A-5D) For a diagonal three-pixel pattern that starts and ends at the same pixel, the match process looks in both the X and the Y directions for possible conflicts. (Figures 5E-5H) In Figure 9A, if any of **A**, **B**, **C**, **D**, or **E** are black then consider moving 2 beside the start pixel in the direction away from the conflict. If only **A** is black, then there are two choices to be considered for moving 2, horizontally towards **C** or vertically towards **E**.

For a three-pixel pattern where the end pixel is corner-to-corner with the start pixel (Figures 6A-6D), if any of the three pixels away from 1 and 3 touching 2 is black, then mark 2 for deletion. In Figure 9B if any of **A**, **B**, or **C** are black, then 2 will be marked for deletion.

Choice of Move versus Delete

If the match method calls for moving one pixel beside another pixel, then the pixel at the destination of the move and the three pixels touching the destination pixel on the side away from the neighbor pixel must be white. If they are not, then mark the pixel under consideration for deletion instead of for moving. In the example of Figure 10, if **A** is a candidate for moving to location **C** beside **B** and if any of **C**, **D**, **E**, or **F** are black, then **A** will be marked for deletion instead of for moving to b.

Editing

The edits do not need to be done during the matching, but are preferably delayed until all the matching is completed. This ensures that the set of edits doesn't depend on the arbitrary choice of where to start the matching process, and also allows the edits to be prioritized and made conditional on the effects of previous editing operations.

The priority for edits is (1) moves first, (2) then deletes for non-corner points, and finally (3) deletes for corners. Corner points are three pixel patterns in which the end pixel is in corner-to-corner contact with start pixel (Figure 6). For steps (2) and (3) the deletes are ordered according to the position in character space of the pixel to be deleted; pixels with larger Y coordinates go first, and in case of the same Y, pixels with larger X coordinates go first. One skilled in the art will recognize that other priority schemes will work equally well.

The ordering is important because a deletion is cancelled if previous edits have removed the conflict so that the deletion is no longer needed. For example, in the character "n" of Figure 1A and 1D, deletions would have been recorded for both inner serifs, but only the right serif was deleted because this deletion alone was enough to remove the conflict.

When recording a potential deletion, save both the coordinates of the pixel to be deleted and a vector pointing in the direction of the conflict. Then at the time the deletion becomes the highest priority edit, check using the vector to see if any adjacent pixel in the conflict direction is still set black. If so, go ahead with the delete. If not, the deletion is no longer necessary, so skip it.

These examples illustrate only some methods of practicing the present invention. Numerous examples of each of the configurations illustrated above were tested and the method of this invention was used to give characters with improved legibility. One skilled in the art will recognize and be able to practice additional methods that fall within the

teachings of this invention. The enumeration and matching steps, for instance, can be performed as part of a single process. The matching step can be combined with the editing step. One skilled in the art may choose to enumerate less than the entire character outline, and might, for example, enumerate only those sections of the path which are near horizontal or vertical inflection points, near counter-clockwise inflections, or those sections which are known as likely to improperly close. One skilled in the art will recognize that other pixel patterns besides those in Figures 2-6 can be used to practice the method of this invention. The pattern in Figure 11, for instance, can be used in practicing the teaching of this invention. One skilled in the art will also recognize that one or more subsets of the patterns described herein can be used, and that the priority for ordering pixel deletions and the criteria for choosing to move rather than delete the pixel can be modified. A choice of whether to move a pixel, for instance, can be made contingent on the effects of prior edits or that decision could be postponed until the rest of the character is edited to see if the move can be avoided. One skilled in the art will recognize that side-to-side and corner-to-corner touches may be acceptable in some circumstances and the method of invention can be practiced testing only for side-to-side touches plus corner-to-corner touches in no or under limited circumstances.

One skilled in the art will also recognize that the path of the character can be modified to minimize or delete instances of improper touching. One skilled in the art will recognize that it may be possible to perform some of the steps of this invention using the path itself rather than the pixels enumerated by the first step of this method. For example, one might test the path itself for counter-clockwise features, inflection points or proximity to other features of a character.

Claims

1. In a method of displaying a character on a raster device at relatively low resolution, the improvement of identifying regions of the character that improperly touch other regions of the character and modifying the display of that character to move or delete pixels which decrease legibility of the character.
2. The method of claim 1 wherein each point pixel of a displayed character is identified and deleting or moving point pixels that touch another region of the displayed character.
3. In a method of displaying a character on a raster device at relatively low resolution, describing the character by means of one or more outlines, trac-

ing each outline, enumerating the pixels in which the outline intersects pixel midlines and selecting groups of two, three or four pixels having the characteristic of being pointed including one or two point pixels that represent a character point, and matching and moving or deleting one or more point pixels that touch other parts of the character bitmap.

4. The method of claim 3 wherein the enumeration of pointed groups includes selecting those groups of three or four pixels where the outline passes through a first or start pixel, then through a second or point pixel, which is horizontally, vertically or diagonally adjacent to the start pixel, then through a third or point pixel which is horizontally, vertically or diagonally adjacent to the second pixel and then through a fourth or end pixel, which is either the start pixel or horizontally, vertically or diagonally adjacent to the third pixel and also horizontally or vertically adjacent to the start pixel.

5. The method of claim 3 wherein the enumeration of pointed groups includes selecting those groups of two or three pixels where the outline passes through a first or start pixel, then through a second or point pixel, which is horizontally, vertically, or diagonally adjacent to the start pixel, and then through a third or end pixel, which is either the start pixel or horizontally, vertically or diagonally adjacent to the point pixel and also horizontally, vertically or diagonally adjacent to the start pixel.

6. The method of claim 3 wherein places in the character bitmap are identified where a point pixel touches a pixel from another part of the character which is already displayed in the character bitmap and,

if a point pixel can be moved horizontally or vertically so that it still touches the start or end pixel in a pointed group and the point pixel after moving does not touch any other displayed portion of the character bitmap or

if no such move is possible, deleting the point pixel.

7. A method of claim 3 wherein the entire character bitmap is analyzed before any pixels are moved or deleted, then beginning at an appropriate point in the enumerated path, for point pixels which touch a part of the character bitmap where the character outlines do not touch, to the extent possible moving such point pixels horizontally or vertically so that they are still in contact with the start or end pixel of the related pointed group and do not touch other displayed pixels in the character bitmap, for point pixels which touch an inappropriate part of the character bitmap but which cannot be moved, where such a point pixel occurs in a pointed group having start and end pixels which are either a single pixel or start and end pixels which are horizontally or vertically adjacent, deleting the point

pixel then

for any remaining point pixels which touch an inappropriate portion of the character bitmap, where those point pixels are part of a three-pixel pointed group having a start and end pixel which are diagonally adjacent to each other, deleting the point pixel.

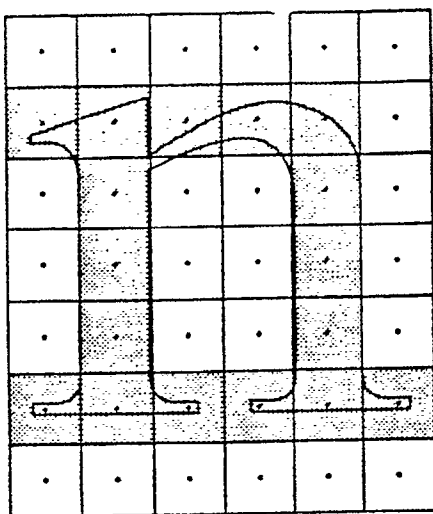


FIGURE 1A

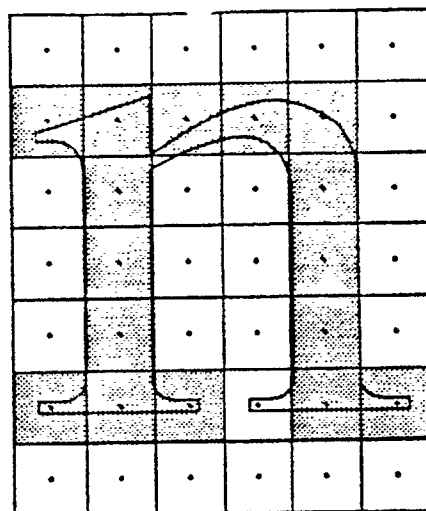


FIGURE 1D

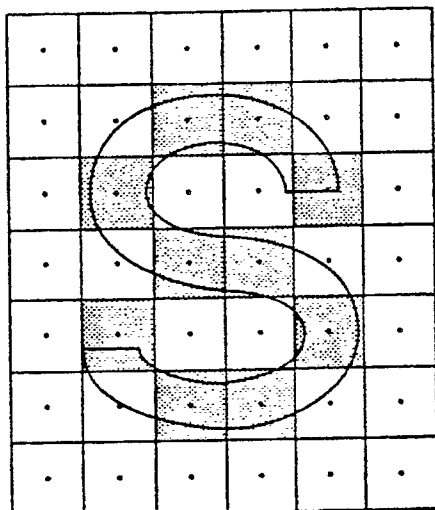


FIGURE 1B

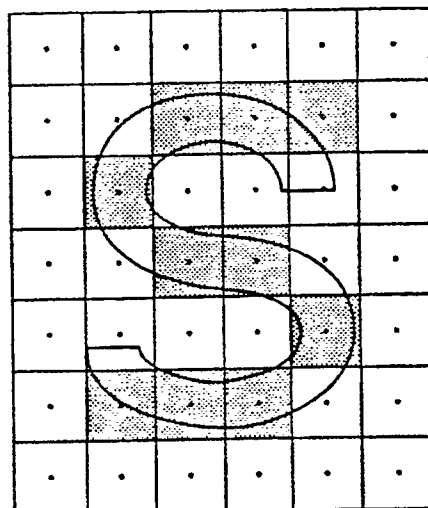


FIGURE 1E

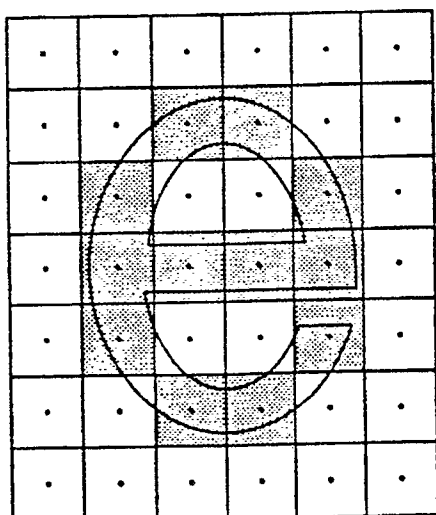


FIGURE 1C

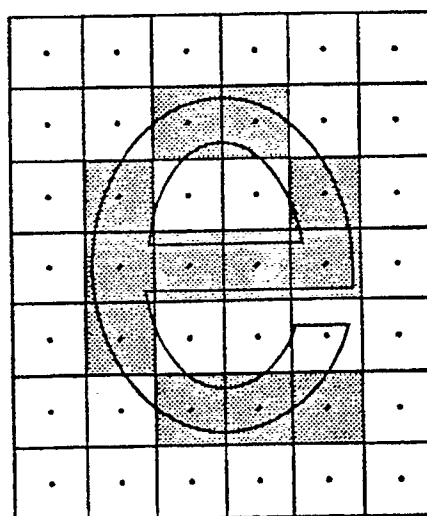


FIGURE 1F

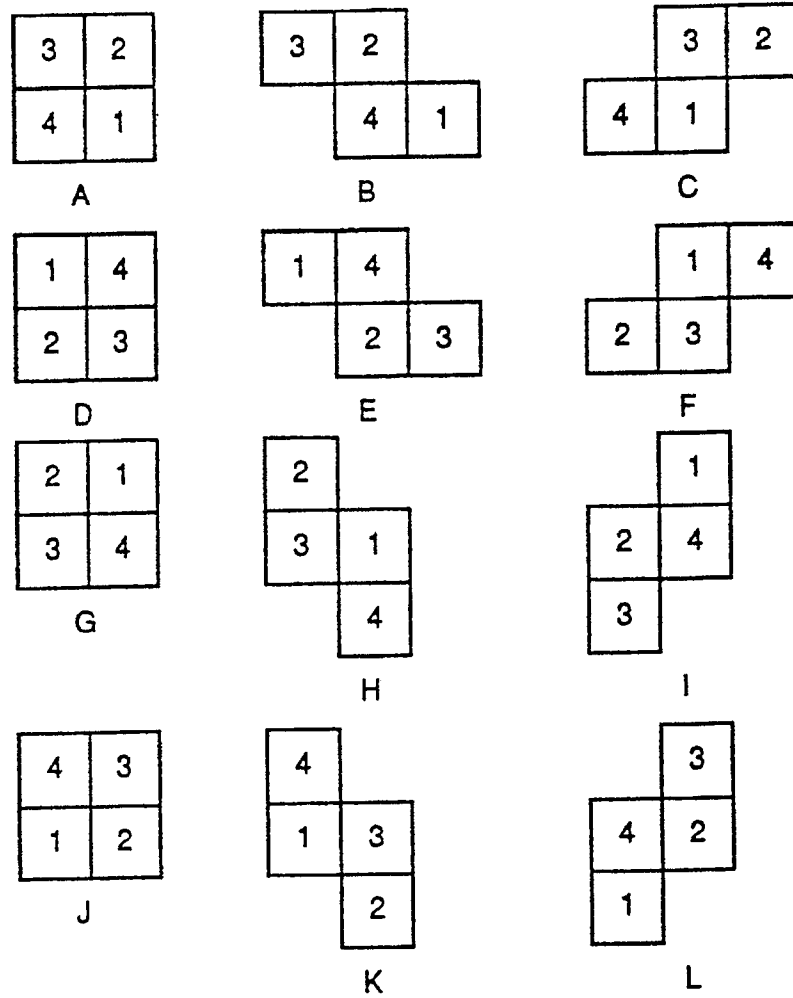


FIGURE 2

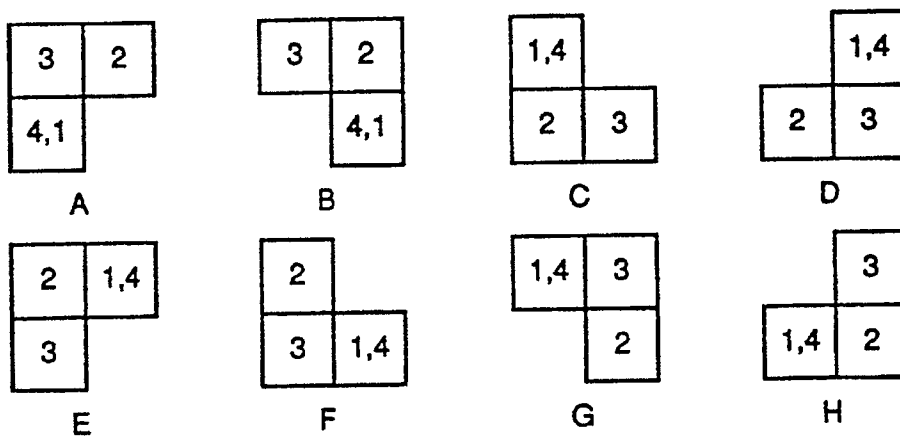


FIGURE 3

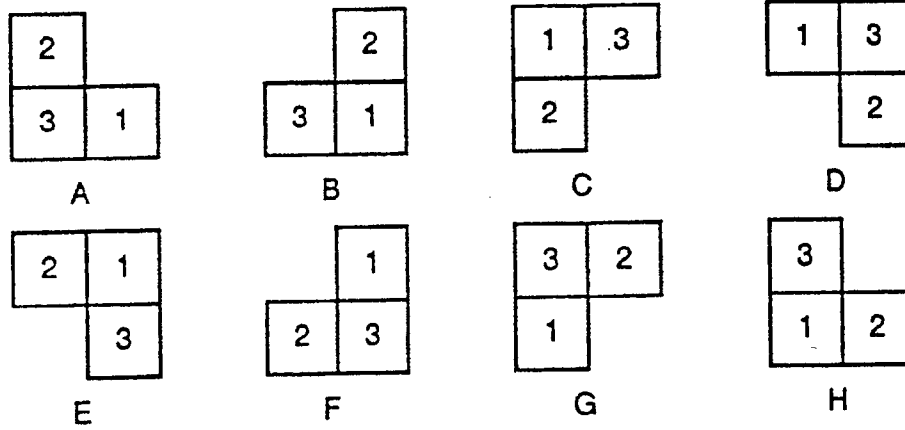


FIGURE 4

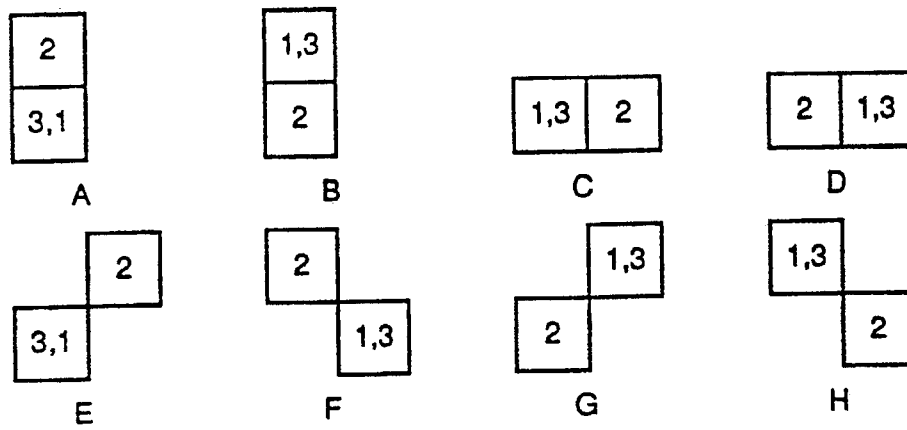


FIGURE 5

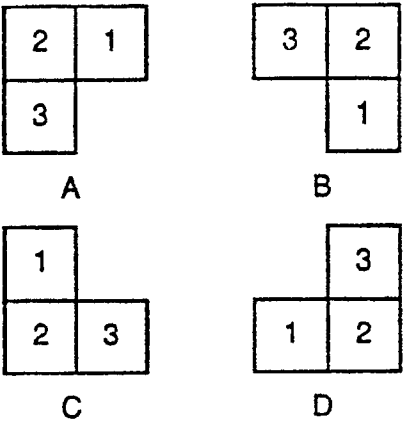


FIGURE 6

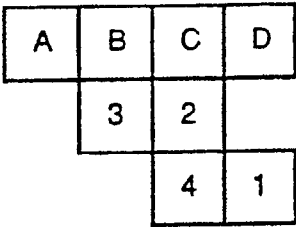


FIGURE 7

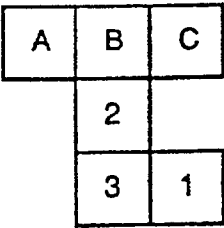


FIGURE 8

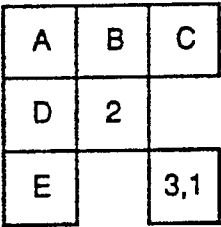


FIGURE 9A

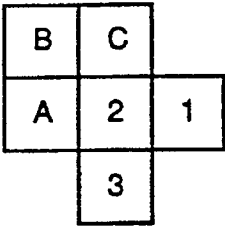


FIGURE 9B

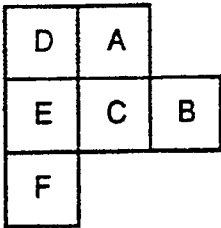


FIGURE 10

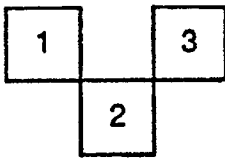


FIGURE 11